NEAR FIELD DENIABLE COMMUNICATION

A Dissertation
Presented to
The Academic Faculty

By

Abhinav Narain

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology

August 2017

NEAR FIELD DENIABLE COMMUNICATION

Approved by:

Dr. Nick Feamster, Advisor
School of Computer Science
Princeton University

Dr. Mostafa Ammar
School of Computer Science
Georgia Institute of Technology

Dr. Taesoo Kim
School of Computer Science
Georgia Institute of Technology

Dr. Hariharan Venkateswaran
School of Computer Science
Georgia Institute of Technology

Dr. Alex Snoeren
Department of Computer Science and Engineering
University of California San Diego

Date Approved: July 10, 2017

To mom and dad, for their love, support and encouragement.

ACKNOWLEDGEMENTS

Many people have contributed to this thesis, either through direct collaboration, feedback, and inspiration, or by providing a supportive environment which has made my life as a graduate student much more rich and productive.

My doctoral advisor, Nick Feamster, has had a tremendous impact on my life over the past six years. He very much leads by example, and his boundless energy, enthusiasm, and optimism has inspired both me and my fellow students. Nick has exemplified all of the characteristics of a great mentor: he helped me select a research topic that both fits my interests and is very relevant to the academic community; he developed contacts with excellent academic researchers both ensuring my work always has an audience and boosting my career prospects; he guided me through the often frustrating process of writing research papers, submitting them for publication, and presenting them for an audience; instilling in me the importance of good communication and ensured I never had to worry about funding.

Alex Snoeren has been an exceptional mentor whose thought process and invaluable time in the beginning of my graduate school has immensely helped me in developing critical thinking as a researcher. His questions have almost always left me searching for more. Other members of my thesis committee, Mostafa Ammar and Taesoo Kim gave excellent feedback and a fresh perspective on the work in this dissertation. My heartfelt gratitude to Mostafa for being very supportive in my early years of graduate studies. Venkat has been always supportive of me in my entire graduate life and helped me in every way while my stay away from Georgia Tech, through emails or phone calls. I am equally grateful to Late Karsten Schwann, who have always smiled at me in hallways and taught an excellent course on Advanced Operating Systems. Jim Xu for some of the most profound advice when I most needed it. I had the fortune to interact with Mung Chiang and Kyle Jamieson, both of them being extremely smart people and shared insights in my final graduate years. I have not bumped into any faculty as many times as Prateek Mittal in E-Quad and al-

ways ended up having some interesting feedback on my research. Although all of them have been busy, they have been quite considerate to share some time with me and I am immensely thankful for their consideration.

I have been fortunate to work with a wide array of supportive, intelligent, and wise collaborators who have helped me learn new areas of science which I could only dream to have understood. Matthieu Bloch for introducing me to Information theory and Coding theory. I am highly indebted to him for his time and patience with me as a graduate student and also being very light-hearted and friendly on many occasions. I hold the same regards for Aveek Dutta for helping me dealing with aspects of electrical engineering, which I was always intrigued by but never had the strength to understand all by myself. Elaine Shi with whose boundless energy and enthusiasm motivated me to work harder. I am grateful to Aaron Schulman and Dave Levin for being special people who have beared with my vexing question and had great research conversations unlike no one else. Faculty at University of Maryland, College Park had been very warm and I thank them for their hospitality during my second year of graduate school.

I have worked my way in my research projects by interacting with many folks on online freenode IRCs, some of whom I have happened to meet in person in some conference or internship. They have been very generous in sharing and extending my knowledge of the platforms I used in my research. Adrian Chad (Free BSD maintainer), Felix Fietkau (OpenWrt/Atheros), Johannes Berg (Linux Kernel Wireless subsystem maintainer) and folks at GNU Radio dev team to name a few while others might just be nicknames on IRCs.

My years at Georgia Tech would be a lot less fun had I not had the pleasure of sharing it with some amazing colleagues. I am greatly indebted to my senior labmates Anirudh Ramachandran, Murtaza Motiwala, Robert Lychev and Vytautas Valancius who have guided me in difficult times and kept my spirits high. They have guided me in every walk of life. NTG lab members have been an inevitable and joyful part of graduate life. Joon Kim, Sam Burnett, Mohammad Shahbaz, Sarthak Grover, Samantha Lo, Maria Konte, Ilias, Arpit

Gupta, Yogesh Mundada, Bilal Anwar, Srikanth Sundarasen have provided insightful and honest feedback on papers and presentations, commiserating on life as graduate students. CS 7001 ties with Dipanjan SenGupta and Ketan Bharadwaj for extended conversations about Operating Systems and Computer Architecture and for being patient ears in difficult times. Special thanks to Keerthi Suria for my time in coding theory class and his suggestions about research.

At Princeton University, Florian Sprung for wonderful discussions and music concerts in Fine Hall, Department of Mathematics. Chinmay Khandekar and Aditya Bilal for long philosophical discussions during hikes in Princeton. Professor Manjul Bhargav (from Dept. of Mathematics) for his great presence and inspirational thoughts in house parties. Enrico Sassoni, Antonio Perazzo, Sonushka Naidu and other folks for introducing me to new things in Princeton. Roya Ensafi, Aylin Caliskan, Phillip Winter, Paul Ellenbogen, Marcela Melara, Laura Roberts, Hooman Mohajeri, Ben Jones from Noise Lab and Princeton Security group for stimulating weekly research meetings. Felix Huang, Bharath Balasubramaniam, Liang Zhang, Yixin Sun, Shirley Wang, Micheal Wang, Jiasi Chen, Carlee Joe-Wang, Maria, Aakanksha Singh and everyone else from Princeton Edge Lab for great lunchtime and hallway conversations. Alessio Piccolo, Cristina Florea, Helena Benveniste, Rachel Bergmann, Isabel Ballan at Princeton with whom I have spent a wonderful time. Vibha and other Princeton undergraduates for great Indian festivals/events. Friends who have enriched my thoughts in Literature and History through the wonderful house get-togethers with Will Schultz, Flori Pierri, Alexis Siemon, Kelsey Ockert and Bingyu Zheng.

I have been blessed with light-hearted housemates over the years who have helped me to stay escape monotonicity of my life now and then like Mishra, Alikaih, Eric, Grace. All my house mates from France (Georgia Tech Lorraine) in Atlanta - Pierre Alexander, Haikel Balti, Jimmy Da Silva, Hamza Chakir, Alex Gth, Camila Apablaza, Fiona Soudan, Rore Ngue, Chloe Osina, for being wonderful. Fredrick Lepoutre with whom I played some of the most intense music, concert buddy and inspired me to get more involved with sound

engineering.

My close friend, Srinivas Narayana who is a constant source of perspective, inspiration, advice and humor. Anirudh Sivaraman at MIT who has also been a profound bud from undergraduate years whose brief but impactful perspective about life have given me a lot of hope. Vimal Kumar from Stanford University, who's been very patient in conversation in my initial years of graduate school and continues to be. Thanks are also due to my mates from Narmada Hostel and the third wing for all the good times and memories.

Finally, I share my past, present, and future successes with my mother Dr. Geeta Narayan, who has been the greatest source of inspiration and strength in my life and of course my father Dr. Ram Narayan, whom I owe all my strong will. My sister, Shivangi who continue to be to most unconditionally supportive, nurturing, and loving family I could ever hope for. Special thanks to my maternal aunts, Tripti and Pushpa, in India who have showered blessings and prayed for my health and studies with all their heart. I am most fortunate to have the safety net of such a wonderful family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

There is an increasing interest of companies and government agencies to snoop on people's daily lives the increasing difficulty for people to handle such scenarios. The need for private communications is perhaps greater than ever before. Government officials have stated that, "if you have enough meta-data you don't really need content" and that, "we kill people based on meta-data". People have long needed to keep the communications among themselves private, but, increasingly, they may want to conceal not only the messages that they exchange, but also with whom they are communicating—or even the fact that they are communicating at all. This latter type of communication is said to be not only confidential and anonymous but also deniable, in the sense that despite exchanging messages, participants can plausibly deny that any such exchanges ever took place.

This dissertation develops techniques and systems that empower users in physical proximity to have mechanisms for deniable communications. Our work builds from the observation of noise in the surrounding technologies like wireless networks or powerline networks. We use noise instead of protocol obfuscation to create deniable channels between individuals who do not want any third party to recognize that there is possible communication in progress. First, we develop Denali, which uses link layer of 802.11 protocol which achieves this by leveraging the weakness of packet corruption in wireless networks due to its ubiquitous nature of being broadcast medium. Second, we leverage innocuous-looking powerline networks used for powering devices in building infrastructure. We build Powerline Whisperer, where one uses physical layer for deniable communication. It depends on the thermal noise and the electromagnetic interference due to devices present in the medium for message cover. Both these systems allow the users to do point-to-point communication and defend against powerful adversaries who might be interested in snooping on the message exchange for malicious reasons.

CHAPTER 1
INTRODUCTION

There are variety of anonymous electronic communications systems have emerged to provide important—and often widely used—communications channels, but most focus primarily on wide-area communications (*e.g.*, Tor [1], which supports communications between Internet-connected end hosts that are often separated by great distances) where deniability can sometimes be provided by hiding in a very large crowd of Internet citizens. In the circumstances we consider, a wide-area anonymous communication system not only introduces unnecessary complexity and latency, but exposes the parties to additional risk by requiring them to send their messages over the wide-area Internet. Tor provides its users with anonymity as one can find that you are using Tor, but they cannot find who you are communicating with. Tor cannot be termed as a deniable communication channel, however it provides deniability with pluggable transport.

Communication has tremendously transformed over the past 25 years. Link layer technologies such as wireless local-area networks (Wi-Fi) and cellular networks now dominate the communication paradigm. Although the technologies at the application layer have changed in the past couple of decades (even the world-wide-web), there are more accessible and open to understanding of people. There have been side-channels developed at the application layer exploiting using protocol obfuscation and side channels using transport layer protocols, much of the lower layers have not been effectively exploited in terms of what they can provide to current people which is not just basic requirement of communication, but also much needed privacy.

People have long needed to keep the communications among themselves private, but, increasingly, they may want to conceal not only the messages that they exchange, but also with whom they are communicating—or even the fact that they are communicating at all.

and anonymous, but also deniable, in the sense that despite exchanging messages, participants can plausibly deny that any such exchanges ever took place. Deniability in wide area networks is offered by Tor to an extent. It can be effective against local adversaries on the Internet but can be detected by global adversary attacks on BGP [2]. Assuming there is widespread surveillance on the Internet, one cannot be sure of the anonymity.

We consider scenarios where people congregate in common public spaces and want to communicate with others in that same space, yet wish to keep their communications both confidential and deniable. Moreover, it may be the case that one or more parties to the communication wish to remain anonymous. Consider, for example, a covert message exchange between a spy and her handler in a coffee shop, a whistle blower in an office environment, or a group of activists who wish to covertly organize a public protest. These scenarios require local communication that is confidential, anonymous, and deniable.

It is meaningful to investigate alternate technologies to be used in such scenarios, some of which might be more applicable in this context. In this dissertation I present systems on two such technologies –802.11 wireless networks which are very popular currently today and powerline networks which are slowly becoming popular for communication between devices. Such technologies are not explored in the past and provide fresh perspective on their usage.

There are different approaches for hiding messages, one being usage of protocol obfuscation, where the mechanism is to mimic the traffic pattern of popular Internet protocols and evade network adversaries measuring traffic data patterns using the Internet measurement infrastructure. On the other hand, we use noise, indicative of natural physical phenomenon and hide covert messages in it. While "noise" might be used in body of research as something which is irrelevant to the actual information (the signal), but we use the term in a literal sense. In specific case of two works – in *DenaLi*, it refers to corrupted 802.11 wireless packets and in case of *PowerLine Whisperer*, it refers to the electrical noise present on powerline channel due to different electronic devices and the thermal noise present as

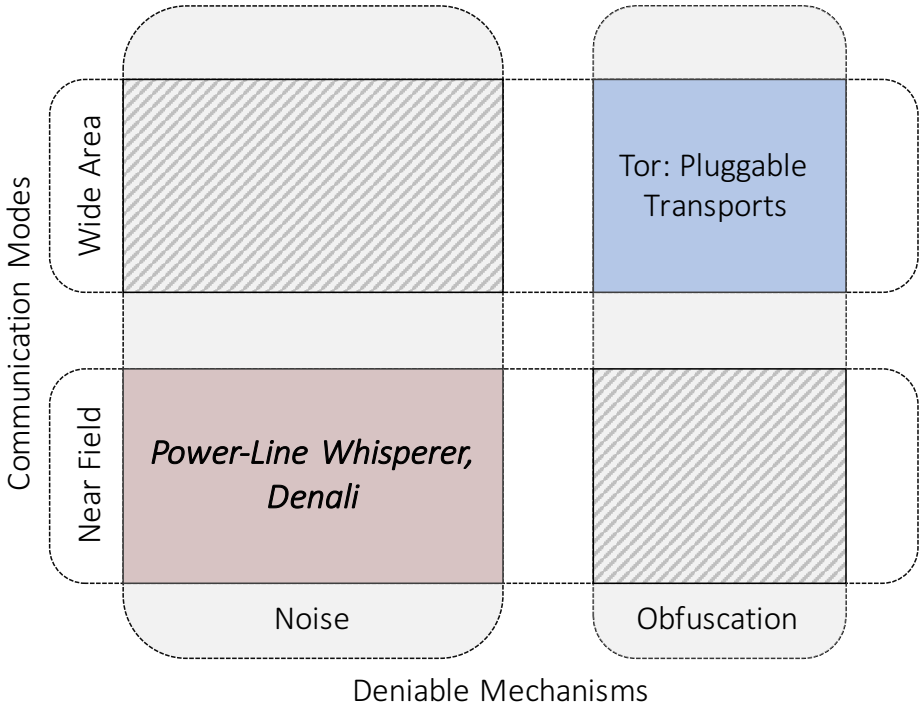


Figure 1.1: Research contribution in broader context

background noise on the channel.

Figure 1.1 puts the dissertation in context with broader research. There are different covert communication channels developed in the past, an extensive and most relevant overview in the context of the dissertation is presented in chapter 2 on Background Work. There are different application layer channels built on top of browser applications such as Tor, which are used by the Internet users primarily for hiding their identity while browsing different internet websites and have certain features called pluggable transports which can be used to deniable communication properties.

For instance, Scramble-suit [3], SkypeMorph [4], meek, obfs4 are application layer deniable communication system using protocol obfuscation operating over Tor bridges over the Internet. Apart from SkypeMorph, there have been covert channels using TCP header fields which as side channels for communication which are not strictly in the umbrella of deniable communication.

The higher the protocol in the networking stack, the longer the distance over which

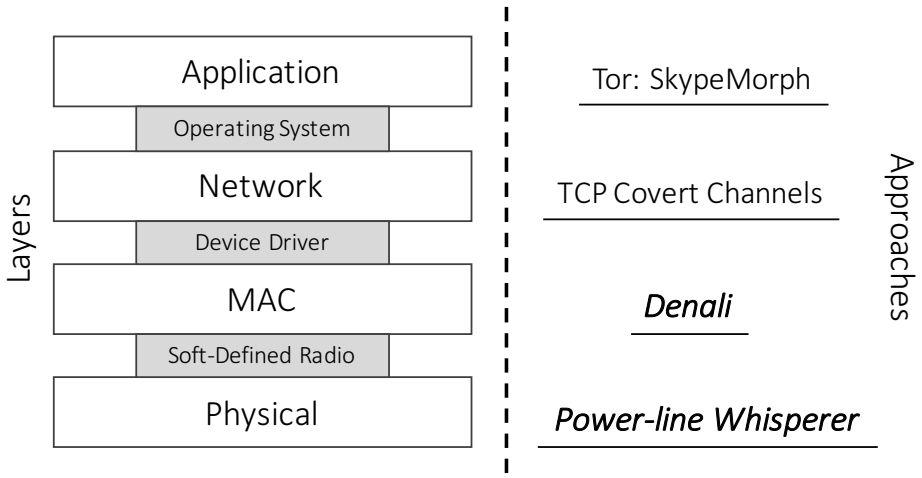


Figure 1.2: Covert channels at different layers of the networking stack for modern communication systems

covert communication systems can be used. Moving lower down the network stack of protocols used by modern communicating devices such as laptops and other static computing devices such as desktops, we witness link-layer protocols. These protocol (such as Ethernet or 802.11) headers are dropped at the first hop of the network and hence can be used only in the case when the intended receiver is the first hop of the network. The Ethernet protocol is implemented for the wired medium which use co-axial cables and well insulated Ethernet wires which are built especially to provide very low packet error rates. On the other hand, the world has continually moved towards wireless networks in the past couple of decades which has given rise to different protocols in the family of IEEE 802.11. These protocols have been evolving to improve with time to provide higher throughput and have battled natural phenomenon of packet corruption in practical wireless channels.

Similarly, leveraging randomness in physical layer channel conditions in characteristics of physical layer lends them helpful to users in physical proximity. These approaches are useful with a view to avoid and bypass adversaries on the Internet which can correlate the user identity with different traffic patterns even in the presence of anonymous communication system as Tor. A state-level adversary lacks a global telescope to identify the activity of an individual with as much as ease as on the Internet and might have to install mas-

sive infrastructure to detect such communication anywhere. It would need several vantage points and much more data collection and analysis that it might not be practical to deploy it.

1.1 Thesis statement

In this dissertation, we posit that there is a need for a different set of tools which take advantages of using local networks such as Wifi and Power-lines to perform deniable message exchange. We use the presence of the ubiquitous phenomenon of packet corruption or channel noise as cover traffic and demonstrate how we can achieve this goal. We develop prototype systems that exploits the presence of noise in different mediums for point-to-point deniable communication.

1.2 Contributions

This dissertation makes the following contributions in defense of the thesis statement:

1. A deniable communication system, *DenaLi* on wireless channel using packet corruption in 802.11 networks at the link layer

2. A deniable communication system, *PowerLine Whisperer* on powerline channel using the ambient noise at the physical layer

Figure 1.2 shows the contribution of the thesis with respect to the networking stack used in modern communication systems. We extend the dimension of deniable covert channels in the direction of different network communication layers, which existed for application layers we now have mechanisms for link layer and physical layer.

Covert channels might appear as a by-product of side-channel. In our exploration, we noticed that noise generated from SMPS regulator of a laptop adapter might be a side channel information which can be used as a covert channel for communicating information

in near-fields. Such phenomenon has been noticed in electromagnetic spectrum in the air as the medium, it has not been explicitly noticed in powerline medium. The primary cause of this phenomenon in mediums such as air and powerline is the same but there is a subtle difference.

The difference in the EMI noticed by using a magnetic loop are directly due to the movement in the power-cycle of the different physical units being powered by smaller power converters on the devices like desktop/laptop. On the other hand, such phenomenon on powerline is a reflection of a second-order activity of the SMPS transformer which changes as a result of combined activity of different physical units performing different tasks simultaneously. This is usually centered at the switching frequency of the SMPS transformer. Due to this limitation, it is not practical to get information about every task executed by the physical units at a finer granularity.
We now elaborate on these contributions.

Recognizing the need for anonymous, deniable communications. There have been communication tools which use the public infrastructure such as telephone and cellular networks, while also the Internet. Such infrastructure takes longer routes passing through multiple hops physically while also logged by different services used by users to make the connection. Instead, in settings where parties are physically close to one another requires for a new class of communications tools. These tools can bypass the use of such massive infrastructure and also various mechanisms to log such communications. We specifically focus on making such means possible. We define the notion of deniability and design modulation schemes that achieve deniability by matching the properties of corrupted Wifi packets to the cover erroneous packets or noise properties of the channel.

Ubiquitous presence of different technologies and phenomenon. There are ubiquitous technologies such as WiFi and Power-lines present already in urban settings which can serve as useful channels for different forms of communication which might provide re-

lief to the growing obsession of capturing every form of information regarding subjects by different corporations and governments. We have WiFi communication (and the corresponding wireless frame corruption) can serve as useful cover to conceal communications. Similarly, presence of noise in powerlines is also an opportunity to leverage it for communicating messages between individuals.

Prototype systems. There has been previous discussions and mention of such communication paradigm in Computer science literature but there have been no real-world implementation of such techniques. We implement prototype systems based on designs in our works and evaluate their performance in different settings. We have some notion of ground truth in such settings with our experiments which we try to match.

The following are some general comments based on the experiences and beliefs of the author rather than claims backed by evidence.

1. *Tradeoff between deniability and throughput.* While building two systems and have revisited an old lesson of the trade-off between throughput and deniability. Sometimes perfect security or privacy is not as important as the usability and portability of the equipment. With different approaches to side-channel, it is difficult to say what is the optimal detection strategy for a technique. There might not exist one in certain cases or might not be worth the time to understand the details. What is practical might not require a detail scientific scrutiny due to the nature of its operation in conditions in the wild. On the other hand, a rigorous scientific strategy for providing deniability might not be a practical solution to be deployed in the wild.

2. *Protocol Obfuscation vs Noise.* Understanding the underlying distribution (corresponding to null the hypothesis of no covert communication) has been the general guiding principle for covert channels. This might look very different when studied in usage patterns of an application, or transport protocols on the Internet to the link-layer and physical-layer work presented in the thesis. The view presented in the

thesis is morphed by digital world and one might apply similar principles to communicate in analog world to achieve deniability. On a cursory thought, one might think using protocol(which can be seen as a strict set of rules) obfuscation might be a difficult task for the construction of covert channel than using the physical layer but our experience has given us the impression that physical layer has it own set of physical laws which make it a challenge to construct covert channel. In one sense, digital world is much more forgiving than analog.

3. *Analytical solutions in real-world setting.* The difficulty with mathematical modeling is the problem formulation assumes no understanding of engineering. Strict Provability of deniability vs practicality of a system can be two requirements which are hard to combine. When we tie ourselves to mathematical models, we have certain assumptions. These mathematical models might not map to the real world and might require certain details which are interesting but do not have an analytical solution. Analytical solutions might be hard to compute or even formulate mathematically. There can be limits that are imposed on models (with notations suggesting values of the random variable, such as deniability tending to zero) which might not be practical. Such limits, in theory, might not transform to algorithms which work in real communication channel with noise.

4. *Theory and Practice.* Engineers often across the problem of clocks for communication systems [5] for synchronization. Such drifts cause the issues of packet corruption. Synchronization algorithms interpolate on the energy on the channel to avoid clock drift and hence the bits are usually scrambled before transmission on the channel. Such requirement might not be fulfilled in the case of sparse transmissions of bits. It might be a challenge to design stealth codes to be used on the channel while keeping the signal close to the noise floor.

As we discuss deniability in different settings in chapter 2, we can say that there is no

correct answer for a perfectly deniable system which works in all the practical settings.

1.3 Outline

We present background on anonymous, deniable and privacy-preserving technologies in Chapter 2. Chapter 3 presents a unified adversary model for the following chapters which is modified later according to practical adversarial capabilities in different settings. Chapter 4 presents a new technique of using packet corruption in 802.11 for deniable communication. Chapter 5 presents a system *PowerLine Whisperer* using powerline channel for deniable communication. We do not exploit an existing protocol standard as in the previous chapter but evaluate a more general technique for covert communication in presence of power-line noise. We conclude with a discussion of general lessons learned while designing these two systems in Chapter 6.

CHAPTER 2
BACKGROUND AND RELATED WORK

2.1 Wide-Area Communication

Many existing anonymous communications [6, 7, 8] systems aim to provide various levels of anonymity in the wide area. One of the most widely used anonymous communications systems is Tor [1], which allows communicating parties to establish anonymous communications channels via a layered encryption technique called onion routing [9]. Users of Tor establish circuits to communicate with each other anonymously in the wide-area. Tor provides anonymity but not deniability, in the sense that users of Tor can conceal who they are talking to, but not the fact that they are communicating using Tor (in fact, Tor is blocked in many countries outright). Tor is not strictly deniable, however, it provides deniability with the use of pluggable transport [10, 3, 11]. There are other systems [12] which consider a different adversary model where the adversary has access to servers (all but one) which can provide deniability between two parties connected to it. Such systems are similar to Tor but explicitly provide a point-to-point communication channel. The fact that one connects to such service might cause suspicion to an adversary and might be a limitation with respect to certain point-of-view. Such systems consider the uncorrupted frame (which have correct forward error checksums) which do not belong to the actual sender as noise. The analysis framework used is Differential Privacy. While in our works, the noise refers to frames which are actually malformed having an erroneous forward error checksum or raw channel noise.

Deniability offered by Tor can be effective against local adversaries on the Internet but can be detected by global adversary attacks on BGP [2]. The focus of this dissertation is different than Tor's: it aims to enable anonymous and deniable communication in settings

where the communicating parties are physically close to one another.

DenaLi bears similarity to other censorship circumvention systems that aim to achieve deniability and covertness in addition to confidentiality and anonymity. Two such systems that operate from end systems are Infranet [13] and Collage [14]. These systems allow participants to establish communications under the cover of innocuous Web traffic: the censor only sees Web requests that are statistically indistinguishable from normal user behavior, thus providing the user with an important degree of deniability, in addition to confidentiality. Other recent systems such as Telex [8], Cirripede [15], and Decoy Routing [16] aim to achieve similar levels of deniability by deploying infrastructure in the core of the network rather than at end systems.

Briar [17] provides a secure, point-to-point anonymous encrypted communications channel between users' devices; like *DenaLi*, Briar enables point-to-point communication, but Briar does not provide deniability.

In all the wide-area network settings, there is a notion of packet which carries information of the user. This frame(packet) has an integrity to it and one can trust the contents of the packets. This is very different when one starts to doubt the integrity of the contents of the packet as in the case of *DenaLi* or not even have a notion of packet but actual raw noise at the physical layer in case of *PowerLine Whisperer*. Packets are trusted source of information and the adversaries use techniques such as deep packet inspection to identify different headers and contents on the packet, which changes in our adversary model where the packets are corrupted and the information they carry cannot be trusted or there are electrical impulses on the channel, the nature of which does not seem to appear to be of carrying information.

2.2 Information Theoretic Security and Steganography

Previous work establishes theoretical limits of deniable communication on additive white Gaussian noise (AWGN) channel [18, 19] and binary symmetric channel(BSC) [20]. The

size of the secret key varies from O($\sqrt{n}$) to O($\sqrt{n}\log n$) depending on the knowledge of adversary's channel characteristics. An optical covert channel is analyzed [21], limited to quantum treatment. In contrast, we work with powerlines, using RF spectrum which brings another set of challenges in building the system. We work with relatively cheap and inexpensive instrument to build affordable and practical transmitter and receiver, instead of highly synchronized channels challenging quantum limits. The quasi-classical channel can be approximated with AWGN channel, similar to our case. We are building a system which considers ambient thermal(due to random motion of conducting particles) and appliance noise, which can be approximated as an AWGN channel.

Steganography [22, 23, 24, 25] is a technique used at the *application* layer. This differs from current work which uses *physical* layer of communication. Steganography embeds the secret message in already existing data (such as images, documents termed as covertext) and assumes that stegotext produced is not corrupted by channel noise, which is sufficiently different from current work. The uncorrupted message is also required to be exported from Alice to Bob, which is a significant challenge to overcome, given Willie is observing the channel. Alice can have positive rate Steganography by embedding n bits in O(n) bit covert text using a secret key of size O(n), and Alice can safely embed O($\sqrt{n}\log n$)) bits by modifying O($\sqrt{n}$) symbols out of n symbols in the covertext, using a secret key size of O($\sqrt{n}\log n$) bits. The $\log(n)$ factor is because the channel to Bob is noiseless. The results in Steganography and covert communication look similar as both use the framework of statistical hypothesis testing and the fact that both depend on the property that relative entropy is locally quadratic [26].

2.3 Near-Field Communication

In cellular communication, *spread spectrum* approach spreads the narrow-bandwidth message over a wider bandwidth. Direct Sequence Spread Spectrum (DSSS) [27] reduces the power spectral density of the transmitted message. This is done by using a spreading code

whose most important non-security application is multiple access, used in Code Division Multiple Access (CDMA). The user data is signal is combined by a code (usually binary) sequence and the duration every element in the code is called "chip time". The ratio of the symbol time and the chip time is called the spreading factor. The transmitted signal will occupy a bandwidth that is spreading times the bandwidth of the user data. In the receiver, the receiver signal will again be multiplied by the same sequence which will recover the original user data. Examples of such codes might be Pseudo Noise code, Walsh Hadamard, Gold and Kasami codes. This spreading of the energy over a larger bandwidth might be thought of as providing stealth to the scheme but it is not a property by design, instead, it is an outcome of providing the ability of multi-user access where different users can have different spreading sequences which can be used to decode the message. Frequency Hopping Spread Spectrum (FHSS) [28] also spreads the code by hopping frequencies using a frequency hopping sequence. It could be hard to be detected by an adversary, yet it is not covert in general as the spreading may not be on the order of $\sqrt{n}$ pulses in n channel use. In comparison, the scheme proposed in this work spreads information in time domain. Although DSSS (used in CDMA) allows the information to hide under the noise floor, it is a very well studied technique with numerous algorithms to blindly detect [29, 30, 31] and estimate signal parameters like center frequency, bandwidth and chipping codes. We do not use spread spectrum techniques as they are not built on principles to provide covertness, rather a by-product of the technique. In contrast, the scheme used has well-defined framework and focus on providing covertness.

Authors [32] use 802.11 *WiFi* protocol for a physical layer covert communication channel. We differentiate our work that it does not require background legitimate communication traffic to hide and leverages noise omnipresent in nature. It also provides more comprehensive framework with finer measurements granting flexibility and strength to the adversarial analysis. Covert channels using light sources like video displays [33, 34, 35, 36] or LEDs [37] or RF [38, 39, 40, 41] require direct line-of-sight for exchanging infor-

mation. On the other hand, wireless medium gives more degree of freedom to the adversary in terms of distance and location for detecting the communication.

2.4 Anonymous Communications

Previous work has sketched systems that use corrupted wireless frames to create a covert channel over 802.11 frames [42, 43] but no previous work has moved beyond paper designs. Calhoun *et al.* designed and simulated a covert channel based upon varying the link rate [44]. This work is purely simulation-based and develops neither a working prototype nor a communication protocol for exchanging messages. None of the previous work analyzes deniability in the presence of an adversary that can monitor channel quality. There has not been any work in powerline communication which would be considered close to providing anonymous, deniable or any other forms of advantage to the party using it for communication. The purview of communication on powerlines is discussed in the following subsection.

2.5 Powerline Communication

There is no covert channel on powerlines to the best of our knowledge. There is a large body of work on using powerline for communication for a different frequency ranges [45]. Instead of building powerline coupler [46] for communication channel that can be configured for different frequency. Contrary to the standard communication requirement, we are interested in electromagnetic spectrum which has seemingly more noise to hide the covert communication. There are commercial products which are capable of communication using Ethernet-over-power technology and others that troubleshoot wireless and Internet-of-things [47]. We have not found open-source device drivers or firmware that can be modified for covert communication. We believe open-sourcing such hardware will allow another set of operational frequencies to be used for deniable communication against the traffic generated by these devices. We will leave it as a future work to be explored in subsequent years

when the eco-system will be evolved.

CHAPTER 3

ADVERSARY MODEL

The aim of an adversary is to detect covert communication with high confidence. In this chapter, we give an overview of a generalized adversary model used in the dissertation. We deal with the specific adversary in chapter 4 for *DenaLi* and chapter 5 for *PowerLine Whisperer*.

In the model shown in Figure 3.1, Alice and Bob are interested in deniable communication; Willie passively listens on the channel. Let $\mathbf{X}$ represent the random variable which represents symbols transmitted by Alice. The random variable $\mathbf{Y}$, represents the noisy version of $\mathbf{X}$ received by Bob, while $\mathbf{Z}$ represent the observations at the eavesdropper. Furthermore, the conditional probability of observation of sequence $\mathbf{Y}$, given the sequence $\mathbf{X}$ was transmitted is $\mathbf{W}_{Y|X}$ and that of the observation $\mathbf{Z}$ at Willie, given sequence $\mathbf{X}$ is given by $\mathbf{W}_{Z|X}$. Alice transmits a message $\mathbf{W}$, which is estimated by the intended receiver, Bob as $\hat{\mathbf{W}}$ with errors due to channel presence of channel noise. The noise power at eavesdropper Willie and Bob are denoted by $\mathbf{N}_w$ and $\mathbf{N}_b$, respectively. The secret key $\mathbf{S}$ is the shared secret between Alice and Bob, unknown to Willie.

Let the adversary use a binary hypothesis detector with the following two hypothesis:

- H_0: Covert communication is *not* present.
- H_1: Covert communication is present.

Let P_0^n denote the n-fold probability distribution of observations measured at Bob. Let Q_0^n denote the n-fold probability distribution of observations measured at Willie, when there is no communication between Alice and Bob over n channel instances. Let $\hat{Q}^n$ denote the n-dimensional probability distribution observed by Willie during actual symbol transmission by Alice. Willie can construct an optimal statistical hypothesis test that min-

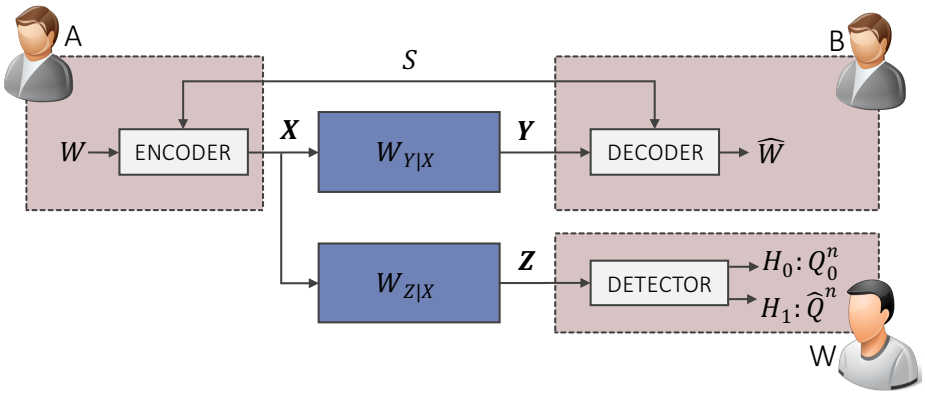


Figure 3.1: Security Model

imizes the sum of error probabilities $\alpha + \beta$. This test yields the tradeoff of probability of error [48] as $\alpha + \beta \geq 1 - V(Q_0^n, \hat{Q}^n)$, where n is the number of uses of the channel. $V(Q_0^n, \hat{Q}^n)$ represents the variational distance (*i.e.*, a measure of difference) between the true distribution Q_0^n because of noise, and the estimated distribution $\hat{Q}^n$ in the presence of communication.

This is a formal framework to model adversary in communication over a channel. We use this framework in different details in the two works in the thesis. In a general sense, to some extent, this approach is used in cryptography where the text is compared in the presence and absence of a good encryption scheme. We use this purely in the context of the message received by adversary over a noisy channel.

The statistical distributions P_0^n and Q_0^n can represent the application layer or natural phenomenon at physical layer depending on the covert channel. The principle remains the same while the treatment may vary significantly. The heart of the argument lies in the difference of the perturbed distribution due to the presence of covert communication and the original distribution. This is captured by the variational distance between the two distribution which can be measured as L1 distance as in the case of *DenaLi* or KL distance as measured in *PowerLine Whisperer*. The limitations of such a theoretical measure are commented on chapter 5 followed by experimental analysis.

Adversary have different capabilities in the two cases of *DenaLi* and *PowerLine Whis-*

perer. *DenaLi* has an adversary which works at the Medium Access Control(MAC) layer, where it is restricted to measurements and distribution over the bits decoded by the wireless ASIC. In *PowerLine Whisperer*, the adversary is much more capable. This comes at the cost of form factor as we prototype the system using software-defined radio. The adversary works at the physical layer and measures electrical voltages from the power-line channel.

The secret key **S** can be thought of an abstraction for the two pieces, which not only includes the previous definition of the location of time slots in the case of *PowerLine Whisperer* but also the time, place of meeting as well as the frequency of operation of transmission of message. It is the information that the adversary does not know. In the case of *DenaLi*, it would represent the time and place as well as the cryptographic keys which would be used for deriving the session key. These would need to be an out-of-band exchange between Alice and Bob.

Unlike a pre-conceived notion of noise being completely unpredictable quantity, noise has statistical properties, depending on the channel. Although the statistical properties can be well defined the interaction of channel with different entities in it might cause sudden and unpredictable events which might not be well captured by the mathematical model. Wireless channel has been researched in depth due to the development of reliable wireless technologies in the past decade and still faces challenges at the physical layer. These problems at the lower layer and the developed inconsistencies are tackled at higher layers of the networking stack. The layered approach provides excellent separation for handling reliability challenges in communication. *DenaLi* build on the MAC layer and hence does not assume a channel model. The physical layer is abstracted by the logic burnt into the ASIC below in the device driver. *PowerLine Whisperer* analysis is built on AWGN channel. This is done due to the simplicity offered by the channel model. Any real channel can be fragmented over a smaller bandwidth which individually resembles AWGN characteristics. We have not undertaken complex models as the analysis becomes analytically cumbersome and does not provide any explicit advantage.

Another dimension to identify the capability of the adversary is physical proximity or the location with respect to the transmitter and the number of instances conducting measurements. Each of the chapters 4, 5 completely define the adversary and present arguments for complex scenarios.

Apart from modeling, one needs to compute the likelihood ratio and evaluate its performance. From an engineering point of view, even though the model is good, one might not know the parameters in it, *e.g.*. covariance function might not be enough to justify numerical evaluation of the formulas from the model. One might think of white noise assumption as a part of engineering contribution. It is important to understand that *white noise* is an idealized realization of noise whose bandwidth is much large than the signal. One might be inclined to study cases of colored noise, that is the noise whose power spectral density varies with frequency. Such a solution is difficult to obtain mathematically and modeling the exact physical phenomenon might also be non-trivial. An example would be a noise who is well behaved such that it is twice differentiable. Modeling noise as white noise is a way of leaving less important details so that one does not have to involve himself in performing tedious operations on data.

The signals to be detected in can be of different types. The first type of signals might be deterministic signals, which have certain parameters that are used to generate them. This would be the case of *PowerLine Whisperer* where the amplitude, phase or the initial phase can be configured by the transmitter. The second type might be stochastic signals, such as in the case of noise generated by electrical appliances as discussed in exploring SMPS noise. Adversary can also be able to model and understand the channel where the parties are interested in performing deniable communication.

3.1 Channel characteristics

In the wide-area network, modeling the channel will depend on the threat model. It will involve the traffic pattern of the interested parties. In the case of deniable communication

using the underlying traffic of Skype, would require to model it as a discrete channel. Depending on the kind of adversary who looks at packet timings or the distribution of bits in the Skype packets using deep packet inspection.

Such deep packet inspection engines are expensive and used by authoritative governments to block access to certain content on the Internet. Although they have not been specifically used for blocking deniable communication but are extensively used to block the underlying services like Tor.

In case of *DenaLi*, the adversary has access to only the link-layer information or in other words modeled as a binary channel. *PowerLine Whisperer* uses the physical layer, which is approximate as an AWGN channel (more digital values than just 0 and 1) and we model information exchange as trigonometric signals in the most fundamental sense.

The general formulations in statistical literature of sequential hypothesis testing and statistical detection theory are used for signal detection problems. As mentioned above, a generic problem is to choose between two hypothesis as with signal having a form as

$$s(t) = \alpha A(t) cos(\omega(t) + \theta)$$

The signal can be of broadly three different forms with different information available to the adversary -

1. Completely known signal
2. Knowing the general form but not the exact parameters of amplitude or phase
3. Stochastic process

PowerLine Whisperer has an adversary which falls in category 1 while there might be adversaries which might want to learn about the noise from devices which might be broadly categorized into category 3 as the noise process is a stochastic process generating some structure in the noise produced with respect to the frequency, intensity and other parameters.

Real world channel can have different characteristics which can mostly be discussed in the context of the kind of noise present on the channel. The channel can have the following kinds of noise

1. Pure white and Gaussian
2. Partially colored and white Gaussian
3. Pure colored

It would not be practical to assume that the noise is white noise as it as an impossible idealization of noise with a bandwidth much greater than that of the signal of interest. On the other hand, using purely colored noise is also not very useful as it renders itself to complex mathematical treatment and also not encountered in everyday processes. Using colored noise in analysis would ask one to incorporate the difficulties present with operating on the data for the inadequate presence of high-frequency components which would require being well behaved. The difficulties reflect themselves in application of integration or differentiation of these noise processes which would require delving into deeper mathematics and might not be that consequential.

In the real world, we have to deal with this aberration in the mathematical models. In our experience with Power-line channel, we find it is a combination of colored noise and white Gaussian noise. The colored nature of the noise primarily shows up due to the presence of various devices which emanate frequency at the switching frequency of the adapters and this spillage of energy is not uniform.

The adversary employed in *PowerLine Whisperer* knows that it is colored noise. The deniability argument for an AWGN channel will still work in this scenario. Borrowing from the proof, the changes in colored noise are due to the presence of high variability in the noise leading to high frequency components. This causes problems in the calculating the higher order derivatives on the model of the data captured from the channel.

The following hint might strictly apply on continuos domain signal, one can think about the intuition from the Taylor's series expansion [20], of the function, that it will not be possible to compute higher order derivatives of the captured data when it is sampled from colored noise. $f(b) = f(a) + f'(a)(b-a) + \ldots + \frac{f^{(n)}(u)}{n!}(b-a)^n + \frac{f^{(n+1)}(\epsilon)}{(n+1)!}(b-a)^{n+1}$

where $f^{(n)(x)}$ denotes the n^{th} derivative of f(x), and ϵ satisfies the relation $a \leq \epsilon \leq b$.

An elegant argument for the proof to apply on non-AWGN channel is to decompose the colored noise channel into narrow frequency ranges which can be approximated to be of constant power spectral density(white) and apply the Square Root Law on the divided frequency ranges.

CHAPTER 4

DENIABLE LIAISONS

4.1 Introduction

The need for private communications is perhaps greater than ever before. People have long needed to keep the communications among themselves private, but, increasingly, they may want to conceal not only the messages that they exchange, but also with whom they are communicating—or even the fact that they are communicating at all. This latter type of communication is said to be not only confidential and anonymous, but also deniable, in the sense that despite exchanging messages, participants can plausibly deny that any such exchanges ever took place.

In this paper, we consider scenarios where people congregate in common public spaces and want to communicate with others in that same space, yet wish to keep their communications both confidential and deniable. We call such a message exchange a *deniable liaison.* Moreover, it may be the case that one or more parties to the communication wish to remain anonymous. Consider, for example, a covert message exchange between a spy and her handler in a coffee shop, a whistleblower in an office environment, or a group of activists who wish to covertly organize a public protest. These scenarios require local communication that is confidential, anonymous, and deniable.

Covert agents have long employed a wide range of techniques in these scenarios, but they tend to be either limited in bandwidth (*e.g.*, a necessarily brief, clandestine conversation) or interactivity (*e.g.*, a "dead drop" of a physical message or storage device). Indeed, any real-world interaction bears some risk of observation, and most are not readily applicable to broadcast scenarios. Hence, a variety of anonymous electronic communications systems have emerged to provide important—and often widely used—communications chan-

nels, but most focus primarily on wide-area communications (*e.g.*, Tor [1], which supports communications between Internet-connected end hosts that are often separated by great distances) where deniability can sometimes be provided by hiding in a very large crowd of Internet citizens. In the circumstances we consider, a wide-area anonymous communication system not only introduces unnecessary complexity and latency, but exposes the parties to additional risk by requiring them to send their messages over the wide-area Internet. We argue that such schemes are not always available due to the widespread Internet arms race of blocking such services and instead a powerful local scheme can be used which leverages the broadcast nature of wireless communication. We argue that these settings call instead for an anonymous, confidential, deniable communications system for the local area that takes advantage of communications devices that users already own (*e.g.*, laptops, smartphones, tablets), without requiring that covert messages traverse the wide-area Internet.

We introduce *DenaLi*, a lightweight communications system to support deniable liaisons. *DenaLi* makes it possible for parties to exchange messages with one another in a local setting, without ever exposing with whom they are communicating, or even the fact that they communicated with a local party at all. Our system takes advantage of the ubiquitous deployment of 802.11 wireless communications networks and, in particular, the pervasive nature of corrupted frames on these networks. Frame corruption is a common phenomenon that inevitably results from a variety of factors, ranging from colliding transmissions to a noisy communications medium. Traditionally, corrupted frames are viewed as a source of inefficiency, as they require the sender to retransmit the original frame; yet, in our case, they provide an opportunity to hide communications. *DenaLi* creates spurious corrupt frames by injecting covert messages into frames carrying cover traffic directed toward innocuous destinations. Since these frames are indeed corrupt, they will not be forwarded by the access point to their apparent destination. Instead, other nodes in the WiFi network that overhear the frame and posses the appropriate secret key can extract and decrypt the injected payload.

DenaLi is conceptually simple, and achieving anonymity and confidentiality is easy enough—any reasonable encryption technique will suffice. The challenges entail designing the communications channel so that the resulting stream of corrupted frames is deniable, which requires both understanding (and modeling) the properties of bit errors in an 802.11 wireless communications channel and appropriately modeling the attacker. To do so, we build on previous work that studies bit-error characteristics in the wireless medium, and perform our own measurements to understand these error characteristics in various settings and for different encodings. We develop a modified 802.11 wireless driver that modulates the covert message over a stream of cover traffic in such a way that the resulting sequence of corrupted frames mimics the existing pattern of corruption in the wireless channel. *DenaLi* traffic matches naturally occurring wireless corruption both in terms of the frequency of corrupted frames and the bit positions within the frames that are corrupted.

DenaLi provides deniability in a setting where an adversary can observe wireless communications in the local area, but cannot get very close to the suspected sender. An adversary who observes transmissions sufficiently close to the sender could infer the presence of a hidden message channel due to the (relatively) high level of packet corruption near the point of transmission. We envision that in typical cases an adversary would not be targeting an individual sender but would rather only be in a position to monitor a group of users (*e.g.*, in the midst of a larger group, perhaps close to the access point). In these cases, we demonstrate through empirical measurements that distinguishing *DenaLi* transmissions from naturally occurring corrupted wireless frames can be made arbitrarily difficult for message rates that can easily support the exchange of short covert messages. We show through extensive controlled experiments with real wireless chipsets that when we closely match the frame error rate and bit error distributions of the existing wireless channel, *DenaLi* achieves a bit error distribution pattern that is indistinguishable from naturally occurring errors. To achieve this level of deniability, throughput is quite low (sufficient for exchanging only small messages or "tweets"), but the sender can, of course, accept less

deniability in exchange for higher throughput, a tradeoff that we explore in our evaluation. Traffic that the user is already sending as part of normal communication can provide the necessary cover traffic, which means that *DenaLi* does not need to create additional cover traffic but can rather hide its messages in the user's existing traffic.

Our work presents several contributions. First, we recognize that the increasing need for anonymous, deniable communications in settings where parties are physically close to one another calls for a new class of communications tools. Second, we observe that in these settings, the ubiquity of other WiFi communication (and the corresponding wireless frame corruption) can serve as useful cover to conceal communications. Third, we define the notion of deniability in this context and design a modulation scheme that achieves deniability by matching the corruption properties of the deniable messages to that of the cover traffic. Finally, we implement and evaluate a prototype system based on this design.

DenaLi's design is inspired by Rivest's proposal for chaffing and winnowing, whereby a sender disguises the real message intended for the recipient by including additional "chaff" on the same channel [49]. With knowledge of a shared secret, the recipient can identify and discard the chaff, leaving only the message in question. Unlike Rivest, however, we further encrypt the message to make it easier to efficiently inject into the chaff without disturbing the statistical properties of the aggregate.

DenaLi is the first system to provide a point-to-point deniable communication channel in a WiFi network using commodity hardware.

Section 4.2 defines our expected usage scenario and outlines our basic approach, threat model, and design goals. Section 4.3 describes the design of the *DenaLi* communication channel in detail. Section 4.4 describes our prototype implementations and explains the changes we made to the wireless driver to enable *DenaLi*. We evaluate *DenaLi* in Section 4.5, discuss limitations and future work in Section 4.6, and conclude in Section 4.7.

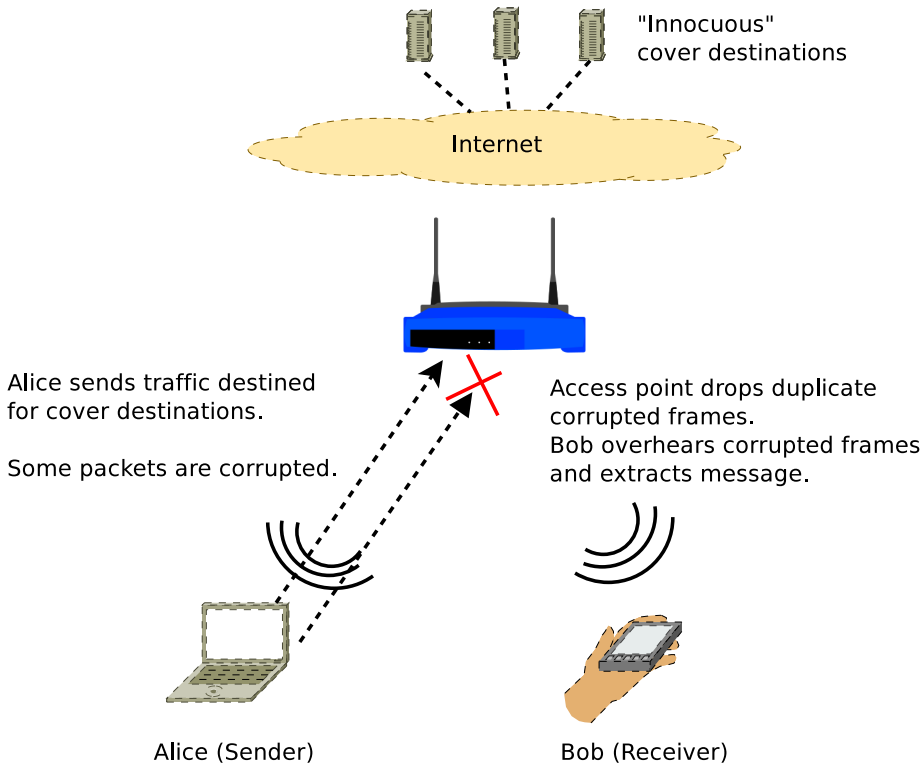


Figure 4.1: Basic communication setup.

4.2 Problem Description

We now explore the scenario where we believe that *DenaLi* is most likely to be used and the threat model, in terms of the capabilities of a typical adversary who might try to discover or thwart communication with *DenaLi*.

4.2.1 Usage Scenario and Basic Approach

DenaLi is designed for settings where the communicating parties are within wireless range of one another and, hence, can hear one another's wireless transmissions to a local access point. We further presume that a *DenaLi* sender has some number of pre-existing connections to innocuous destinations on the Internet which will provide cover traffic for our covert communication channel. Figure 4.1 shows such a basic setup. An adversary may be positioned anywhere in the wireless network and is able to eavesdrop on any transmissions

by the participants. *DenaLi* does not employ or require link-layer encryption schemes (like WEP or WPA) for its confidentiality guarantees.

In this scenario, the sender, Alice, sends traffic to her usual set of wide-area Internet destinations via the access point. Due to the nature of the wireless channel, some frames may experience corruption, and the access point will thus discard those frames. Alice will subsequently retransmit these frames until they are successfully received by the AP and forwarded on. But, if Alice and Bob share a secret, Alice can inject additional, deliberately corrupt frames, such that the frames corrupted by the wireless channel serve as chaff to conceal the fact that some of the corrupt frames contain a hidden message. If Alice and Bob share a secret, Bob can determine which corrupted frames are chaff and can retain only those corrupted frames that contain the hidden message.

Corrupt frames naturally result from various wireless effects, including low signal-to-noise ratio (SNR), broadband interference, hidden terminals, and multi-path fading, which depend on the relative position of the transmitting device and nearby wireless devices, materials of nearby objects, and other unknown factors. Because the causes of corruption are diverse and time-varying, detecting the hidden messages with certainty requires either knowledge of the secret, or the ability to monitor frame corruption rates and compare the measured distribution to the corruption rates that would be expected as a function of both space and time.

To construct a profile that closely matches that of a normal wireless channel, we exploit two important observations about the corruption of packets in a broadcast medium, particularly the 802.11 protocol. First, packet errors in packets occur in chunks of bytes [50], not as individual bits; most of the chunks of errors are about 400 bits, and occurrence of larger chunks of errors is not very usual. This phenomenon might occur as a result of interference, or the loss of synchronization. The second observation is that the bit errors inside wireless frames have specific patterns [51]; for example, bits that are farther from the start of the frame will experience an increasing probability of corruption.

4.2.2 Threat Model

The adversary's primary goal is to detect the presence of hidden communication on a shared wireless medium. If the adversary is able to further determine which transmitted frames contain hidden communication, it may be able to use existing techniques to determine the identity of the sender [52]. The adversary's main capability is to listen to wireless frames within its radio range.

We assume that the adversary has finite computation resources and a finite number of nodes that it can use to monitor the wireless channel. In our prototype, we assume the adversary has only one node with which it can monitor and has knowledge of at least one party which may be communicating using *DenaLi*. In a practical scenario, the adversary might know the identity of the sender but not his MAC address. He would still have to scan the channel and apply techniques to identify the sender's device, which might be hard in dense public places where signal strength of device varies considerably due to the commotion [53]. If the adversary has previous knowledge about which parties may be using *DenaLi* to communicate, it could position its radio(s) close to one of the senders and attempt to determine if the sender's wireless interface was sending corrupted frames at a rate that exceeds the typical rate at which a wireless radio emits corrupted frames. We assume that the adversary remains at a sufficient distance that it cannot conclusively determine that some frames are already corrupt when they are transmitted by the sender; rather, it can only monitor the frame corruption rate. As long as the adversary is sufficiently far away, the sender can always make his channel worse by staying far away from the public access point, thereby legitimately retransmitting at a higher rate than normal.

Even without knowledge of communicating parties, a stronger adversary can monitor and collect wireless transmissions from multiple independent locations in the network and run statistical analysis on the collection of captured traffic. In these cases, the adversary might be able to determine that the profile of bit-error corruption for certain nodes in the network does not match the corruption profile for other senders, or that the frame corrup-

tion profile does not change with increasing distance from the sender as one might expect. Such an adversary might be able to perform an analysis of error patterns with a tool such as Jigsaw [54], but even with the benefit of multiple observation points, if the distributions are matched appropriately, the perturbations that *DenaLi* introduces should still provide deniability for senders. Moreover, a global adversary, *i.e.*, one that can monitor at multiple locations in the wireless network—but not the sender or receiver—does not necessarily have a better chance at detecting the presence of hidden communication than a local adversary who only has one monitoring point. Although the ability to observe transmissions at multiple locations provides the opportunity to observe corruption patterns of the same packet at multiple locations, these observations still do not allow the adversary to ascertain what bit errors would look like at the exact sender and receiver locations [55]. Previous work suggests that bit error patterns within corrupted frames will differ depending on the adversary's location [56].

In our empirical evaluations and security analysis (Section 4.5), we assume an adversary who can observe all the corrupted frames from a single location in the network. We note that even if an adversary targets a particular sender (e.g., based on previous knowledge), the sender can always move away from a suspected adversary or maintain enough mobility to reduce the likelihood of being monitored at close range. (Indeed, previous work shows that simply rotating the communication device can dramatically impact the channel quality [53].) Therefore, we believe that it is extremely unlikely that an adversary could successfully target a sender *and* successfully monitor the sender at close enough range for an extended period of time without tipping off the sender.

4.2.3 Design Goals

We aim to develop a covert channel with a variety of properties, in addition to the standard properties of *confidentiality* and *covertness*. *Undetectability* says that the adversary cannot detect the presence of any messages. *Deniability* is a slightly weaker property that says

that even if the channel is detectable, the adversary cannot determine with non-negligible probability that a particular user or group of users is exchanging messages. *Unlinkability* says that an adversary may be able to detect the presence of communications, but cannot link the sender of a message with its receiver. *Robustness* says that the adversary should not be able to disrupt the channel.

DenaLi technically does not achieve strict undetectability, since the process of sending a message does perturb the wireless channel from its original state. We design the resulting bit error profile to be statistically similar to a normal profile, however, making it difficult for an adversary to determine with certainty that the channel has been perturbed. Because frame corruption is a random process that is itself based on a non-stationary distribution (*e.g.*, it is affected by a variety of factors, ranging from the presence of other senders, to changes in obstructions such as people and doors, to the user's wireless radio, to physical properties of the air), we can perturb the corruption profile of the channel without allowing the adversary to determine that a sender is definitely sending a hidden message. In this way, we achieve deniability.

Independently, *DenaLi* achieves unlinkability because even if the adversary could detect the presence of additional corrupted frames, without having the key that Alice and Bob share, the adversary cannot determine that Bob is the intended recipient of the additional corrupted frames. In fact, by having multiple participants share a group key, *DenaLi* can be used to surreptitiously broadcast a hidden message.

Finally, *DenaLi* achieves practical robustness by virtue of the fact that an adversary cannot easily selectively disrupt the communication of the wireless frames containing the hidden message. An adversary could jam the entire wireless channel, but doing so would disrupt communication for legitimate traffic as well.

DenaLi does not rely on 802.11 encryption standards such as WEP and WPA to achieve confidentiality, as we assume that many adversaries may be powerful enough to either (1) join the channel with a known WEP or WPA2 keys (*e.g.*, in the case where the adversary

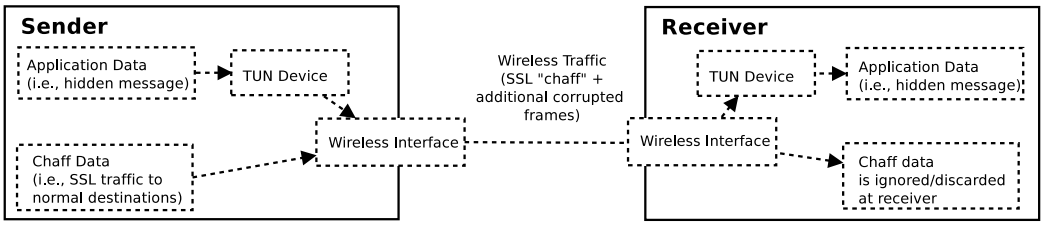


Figure 4.2: Injection of additional corrupted frames via a virtual network interface (implemented as a Linux TUN device).

is the network administrator, such as in a public square or a coffee shop); or (2) break the WEP encryption or WPA2 encryption using known techniques. Instead, *DenaLi* provides confidentiality by encrypting the message contents before injecting them into the corrupted frames.

4.3 Communications Channel

This section describes the *DenaLi* design in more detail. The basic approach is for the sender to inject corrupted frames into an existing encrypted application traffic stream (the chaff), so that in the air, the adversary sees a single stream of encrypted application traffic with non-corrupted and corrupted frames. The goal is to make what is seen on the air appear as a plausible sequence of frames to the purported destination to anyone observing the traffic pattern. To do so, the sender occasionally duplicates existing frames and corrupts them by injecting a portion of the message to be communicated. The sender and receiver must also develop a common means to identify which corrupted frames contain hidden messages, and where (*i.e.*, at what byte offset) within a corrupted frame the hidden message lies.

4.3.1 Basic Mechanism: Frame Injection

DenaLi constructs corrupted frames and hides the corrupted frames among a larger stream of frames being transmitted to the access point. Some of these frames (perhaps including some of *DenaLi*'s constructed frames) will be corrupted by the wireless channel. In order

to make it more difficult to determine which corrupted frames contain embedded messages, *DenaLi* transmits hidden messages only in frames that otherwise are part of encrypted SSL connections (*e.g.*, to popular websites like Gmail). We chose to use SSL connections as the basis for *DenaLi*'s cover traffic because the encrypted payload of these frames acts as a one-time pad into which we can embed similarly encrypted messages without obviously disturbing their statistical properties.

An SSL frame will necessarily have a TCP header, which *DenaLi* uses to compute the offset into the frame at which to place the embedded message. Because bits that are located further into a frame (*i.e.*, with a greater offset) have a greater chance of experiencing corruption [51], *DenaLi* skews the probability distributions on injecting message blocks to favor corrupting bits farther into the frame. Obviously, the message must be (substantially) smaller than the frame into which it is being injected. Our implementation exports a virtual network interface with a small MTU, which ensures that the covert channel is automatically broken into smaller message blocks. Figure 4.2 illustrates the communication tunnel between the sender and receiver, including how the hidden message is combined with chaff before being transmitted over the air; the receiver hears all of the wireless traffic but can discard the chaff before passing the message to the receiver.

Figure 4.3 shows the construction of the combined packet stream in more detail. The hidden message is passed through the virtual network interface (a Linux TUN device), whereupon it is combined with a copy of an existing frame from the chaff via bit-error injection. The corrupted frame is then transmitted very close in time to the unmodified chaff frame. To decode the hidden message, the receiver performs the reverse of this process. Ideally, the entire stream would be transmitted via the same outgoing interface, but limitations of current wireless chipsets prevented us from implementing the transmitter in this fashion; Section 4.4 discusses these limitations in more detail, and Section 4.6 explains how we conceal the presence of two separate transmitting interfaces.

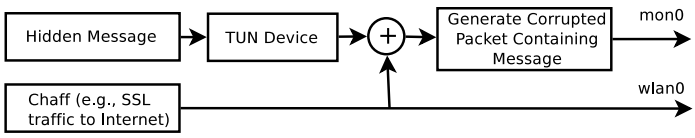


Figure 4.3: Process of injecting corrupted frames at the sender; the receiver performs the reverse of this process.

4.3.2 Communication Protocol

Figure 4.4 shows the steps that are involved in exchanging messages in a two-party message exchange. We now explain these steps in detail.

Establishing a shared session key The sender and receiver use the *DenaLi* channel to establish a shared session key in a manner that is analogous to how session keys are established in many protocols. In case of *DenaLi*, the colluding parties should be aware of that they are in proximity of each other and then instantiate the key exchange process. The sender generates a session key and encrypts the key with the receiver's public key. It then sends the resulting ciphertext over the *DenaLi* channel, taking the resulting ciphertext and embedding it as corrupted bits in an outgoing sequence of frames. The receiver decodes the message from the corrupted frames to retrieve the session key. The session key is transmitted on the *DenaLi* channel just as any other message would be, except that the initial transmission and encoding is based on the receiver's public key, instead of the session key itself. All transmissions on the *DenaLi* channel involve a process of the sender encoding the hidden message and the receiver decoding it upon receipt, as described below.

Encoding and transmitting First, the sender obtains a cover frame by duplicating a frame that is about to go out of its wireless interface as part of an existing connection. It then corrupts this duplicate by injecting data from the covert channel. Before injecting the hidden message into a corrupted frame, the sender: (1) encrypts the hidden message with the shared session key (or, in the case of the initial key exchange, the receiver's public key) using CBC-AES 256-bit symmetric key encryption; (2) computes the offset into the

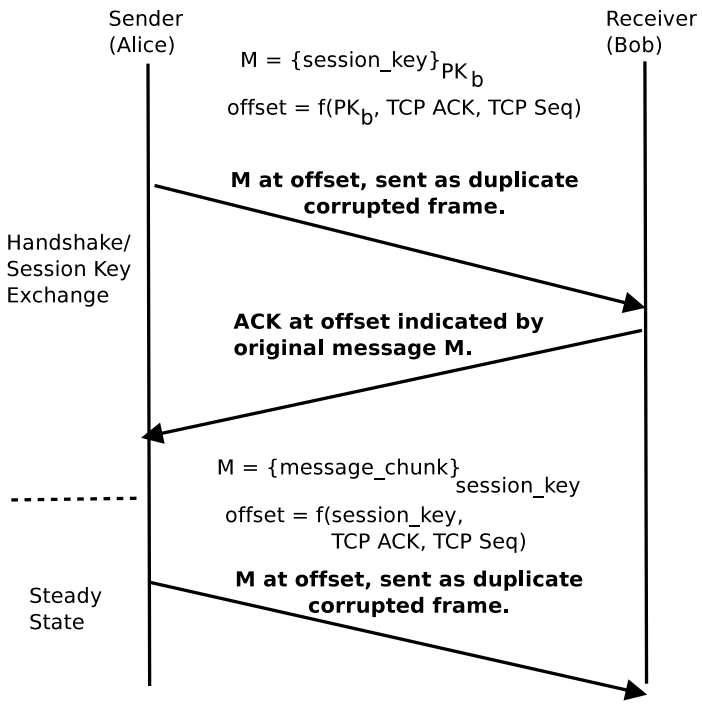


Figure 4.4: Steps involved in exchanging messages using corrupted frames.

frame where the message should be inserted; and (3) computes an HMAC over the message ciphertext. The sender then inserts bits corresponding to the hidden message length, the HMAC, and the hidden message itself as a block into the corrupted frame. We describe the process of computing the frame offset and the HMAC below.

In addition to the session key, the sender uses the TCP sequence number and acknowledgment number as salts to compute the frame offset for the hidden message. Doing so helps randomize the offset, so that the inserted bits are not always in the same location in the corrupted frame; randomizing the offset makes it difficult for an adversary who is eavesdropping to ascertain the presence of a hidden message, since the location of the corrupted bits that contain the hidden message will be different for each packet. We considered using a pseudo-random number generator with an initial seed to allow the sender and receiver to compute this offset; the problem in doing so is that if any corrupted frame containing a hidden message is lost, reordered, or itself corrupted, the receiver and sender will lose synchronization. Instead, *DenaLi* uses the output of a public cryptographic hash function

that uses the TCP sequence number, acknowledgment number, and shared secret (or, in the case of the initial key exchange, the receiver's public key) as the input for computing the offset. Thus, all of the information that the receiver needs to extract the hidden message from the frame is present in the frame itself. Unless the adversary has the shared secret, it cannot determine the offset of the artificially corrupted burst sequence.

Because the injected frame is corrupt (*i.e.*, its layer-two checksum is invalid), the receiver no longer has an inherent way to determine the integrity of the frame—or, more specifically, the embedded *DenaLi* message within—it receives. In lieu of the (now corrupted) frame checksum, a *DenaLi* sender also includes an HMAC computed over the hidden message contents that is keyed on the session key, the TCP sequence number, and the acknowledgment. The message's HMAC is prepended to the hidden message before the resulting bits are inserted into the frame.

The astute reader might observe two nuances about the way that the sender embeds the message into a corrupted frame. First, the message length is included "in the clear". Including the message length in the clear is necessary because the number of bits corresponding to the hidden message varies (both by design to make detection more difficult, and as a natural result of the original message sizes). Because both the value of the message length and the offset within the frame where the bits indicating the message length vary per-frame, recognizing a pattern would be difficult. A sender could, of course, introduce more entropy into the message length value by randomizing the block size for each block that it injects into a corrupted frame, making it essentially impossible to identify the presence of the message length value, at the expense of channel throughput.

Second, all of the corrupted bits are injected into the frame as a single block rather than interspersed at random bit locations throughput the packet. Previous work has established that wireless bit errors tend to occur as corrupted blocks [50], not as individual corrupted bits. Additionally, because the *DenaLi* sender injects ciphertext into other ciphertext (*i.e.*, the SSL stream that serves as the chaff), interspersing the block throughput the packet does

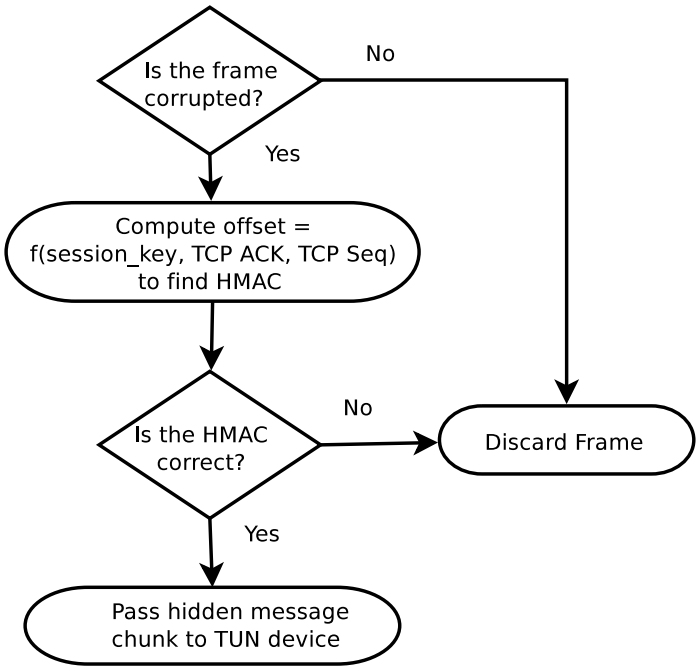


Figure 4.5: Checking the integrity of received hidden messages.

not increase covertness: Because both the hidden message and the chaff are encrypted, the adversary can see that the frame is corrupted, but has no straightforward way of determining the bit positions corresponding to the corruption, unless he has the corresponding uncorrupted version of the frame. Injecting an encrypted message into SSL payload makes the likelihood of every bit to be corrupted to be ≈ 0.5.

Receiving and decoding To receive the hidden message, the receiver polls the wireless medium for all the corrupted frames and attempts to decode and decrypt the bits in each corrupted frame that are located at the appropriate offset, which is computed as a function of both the session key and the TCP sequence number and acknowledgment numbers in the packet header. The receiver can apply the same function to determine the appropriate offset of the message in the corrupted frame to extract the ciphertext and decrypt it to recover the session key, which will be used to encrypt future messages and as an input for computing the frame offsets.

Upon hearing a corrupted frame in the wireless medium, the receiver extracts the grain

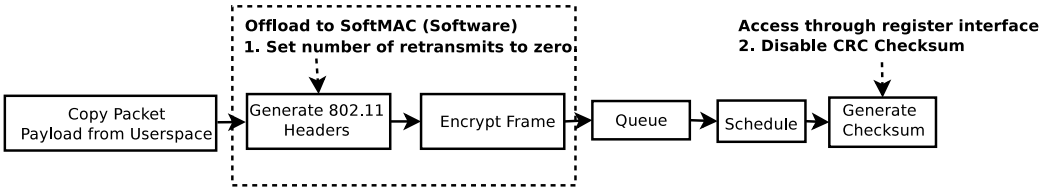


Figure 4.6: Processing of an 802.11 wireless frame at the host, and the two modifications that we make to enable *DenaLi*: (1) setting the number of retransmissions to zero through the SoftMAC implementation; (2) disabling the frame checksum computation to allow the interface to transmit the corrupted frame.

from the chaff by computing the offset where the hidden message is expected (as a function of the key and the TCP sequence and acknowledgment numbers contained in the frame) on every corrupted frame, extracting the bits that should correspond to the hidden message, computing the HMAC on the decoded and decrypted message, and comparing it with the HMAC value present in the packet. The receiver computes the HMAC of the decoded message and compares it to the value of HMAC included in the packet, which (as mentioned above) is prepended to the transmitted message before being injected into the frame. If the HMAC is correct, the receiver then proceeds to decode and decrypt the hidden message. (It is extremely unlikely that the bits of the secret message and the HMAC will be corrupted simultaneously in such a way that the HMAC calculated over the corrupted frame will be the same as the corrupted value of the included HMAC.) Figure 4.5 illustrates this process.

4.4 Prototype Implementation

In this section, we describe a prototype implementation [57] of *DenaLi* using off-the-shelf wireless chipsets based on the design detailed in Section 4.3.

4.4.1 The TUN Interface

In the interest of simplicity, our prototype implementation of *DenaLi* provides a TUN interface that allows applications to use the covert channel just as any other network interface. It is a virtual interface in Linux, implemented as a TUNnel device, to exchange packets with

user space. A user can determine how to design and implement applications that communicate over the channel, or just use existing ones. Once a packet is transmitted on the TUN interface, *DenaLi* encrypts it (including the headers and checksums), calculates the HMAC of the encrypted message, computes the resulting message length, and concatenates them to arrive at the bit sequence that is ultimately inserted into a corrupt wireless frame. We compute the HMAC using SHA-256.

4.4.2 Dual Wireless Interfaces

Most existing wireless chipsets calculate the layer-two checksum, also known as the frame check sequence (FCS), in hardware. Hence, even the "corrupted" frames created by injecting the encrypted payload would normally be sent out with a correct FCS, meaning the destination of the encapsulating chaff frame (*i.e.*, the access point) would receive the packet and attempt to process it. While the IP checksum would still likely be incorrect, it is far less common for an IP checksum to be invalid on purportedly correctly received frames, destroying *DenaLi*'s deniability.

Hence, we must ensure that the corrupted frame is transmitted with an invalid FCS. Unfortunately, the current architectures of most wireless chipsets do not expose an interface to manipulate the FCS. Instead, our prototype uses a wireless interface card with the Atheros AR9485 chipset, which exports a register that disables the calculation of the FCS (we are unaware of other vendors that provide this feature). Atheros *ath9k* and *ath5k* series of chipsets provide this feature available commercially for Linux [58]. The register setting is not selective, however: if enabled, all packets are transmitted without a proper FCS. Hence, in order to transmit the chaff traffic, our prototype employs two wireless interfaces: one to transmit the chaff SSL traffic, and one to transmit the additional corrupted frames that contain the hidden message with a corrupted FCS. We are using two wireless cards to facilitate usability of the prototype, as software defined radios are bulky and hard to carry for general purpose use by non-technical person. We use identical Acer Aspire

One laptops with Intel Celeron processor running at 800 MHz, with Linux 3.2.0 and stable compat-wireless networking stack.

4.4.3 Driver Modifications and SoftMAC

Each wireless frame passes through multiple stages before it is transmitted, many of which occur in hardware by default (and, hence, are inherently challenging to modify), as shown in Figure 4.6. The specific stages depend on the architecture of the particular wireless chipset in use, although we provide a rough general outline that many chipsets follow. First, an application provides the payload to the operating system, which in turn copies the data to driver memory after adding 802.11 MAC header. The driver then encrypts the packet and transmits it; the encryption keys are retained in software, but the encryption process itself occurs in hardware. The transmission control unit manages the fine-grained timing of 802.11, including generating the frame checksum right before transmitting the frame.

Our *DenaLi* prototype makes two changes to the default processing pipeline: it (1) disables the FCS checksum; and (2) disables the retransmission of these frames, which obviously will never generate link-layer acknowledgments. Figure 4.6 illustrates where we made these modifications in the NIC processing pipeline.

To modify the behavior of the wireless interface, we use the SoftMAC 802.11 wireless MAC implementation [59], which offloads many functions of the wireless driver to the kernel subsystem, thus forming a clean interface with various vender drivers and allowing us to modify various parts of the process.

4.5 Security and Performance

In this section, we evaluate the security of *DenaLi* relative to the performance that it achieves. As discussed in Section 4.2, our primary goals for security are deniability and confidentiality, where deniability says that the resulting traffic is statistically indistinguish-

able from network traffic that does not contain any hidden message. We begin with a discussion of the characteristics of the resulting wireless traffic that should appear statistically indistinguishable to an adversary. We then formalize our definition of security in terms of the indistinguishability of the resulting *DenaLi* traffic from ordinary wireless frame corruption.

We conduct real-world experiments with our prototype implementation to explore the tradeoffs between deniability and throughput over the *DenaLi* channel. Across all of our experiments, our prototype consumes an average of 2% and maximum of 5% CPU time at both transmitter and receiver while it injects or decodes corrupted wireless frames. We confirm that no packets are dropped by kernel or socket buffers despite using the pcap library for packet reception and injection.

4.5.1 Traffic Characteristics

Detecting *DenaLi* communication requires the adversary to make observations about perturbations to the natural error patterns at one of two levels: packet errors in the medium, or patterns of bit errors within individual frames.

The *packet error rate* is the fraction of transmitted frames in the wireless medium that are corrupt. This rate depends on the characteristics and nature of the environment where the colluding parties (and the adversary) are located. Although the instantaneous frame error rate cannot be modeled precisely because the type and frequency of events that cause interference or frame loss are inherently random, we can calculate the rate of corruption of the frames in a live capture of a collected packet trace and attempt to mimic that distribution. *DenaLi* users maintain statistics regarding the packet error rate of normal frames so that they can inject corrupted frames in a way that mimics the naturally occurring packet corruption in the current environment. In channels that are subject to corruption rates that are higher or more variable, *DenaLi* participants can inject hidden messages with higher frequency. We explore the relationship between the amount of noise in the channel and the

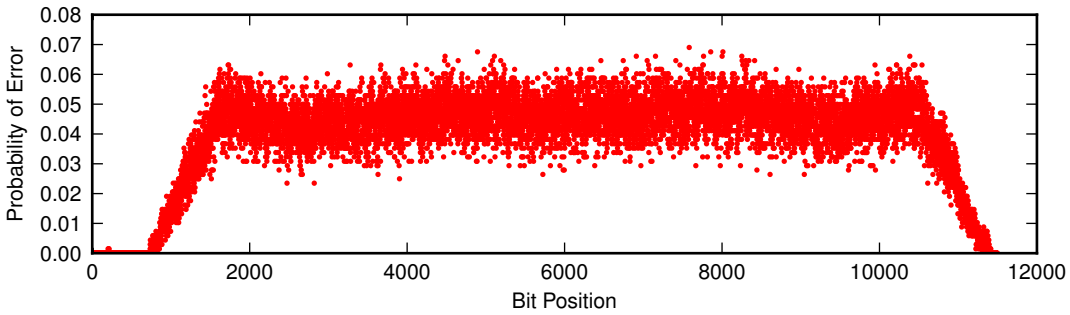
throughput that we can achieve later in Section 4.5.3.

The *bit error distribution* is the distribution of the bit errors in specific positions *within a corrupted frame*. An adversary who captures the frames may analyze the corrupted frames to compare the error patterns. We modify the contents of the intentionally corrupted frame in such a manner that it is difficult to differentiate actually corrupted bits from the crafted corrupted frame. Our goal is to inject bit errors into packets in such a way that the resulting distribution of bit errors resembles a bit-error pattern that would result from the corruption of one or more symbols in an encoded wireless packet. The exact bit-error pattern is difficult to model because these patterns depend on how the sender modulates packets. In lieu of conducting additional experiments on bit error rates ourselves, we follow the assumptions from the Maranello study [50], which suggests that the bit errors in a frame occur in chunks, due to the loss of synchronization between the sender and receiver or the bursty nature of interference in the wireless channel, unlike uniform corruption of bits in the whole frame. In our evaluation, we use *DenaLi* to corrupt specific bit error patterns in such a way that mimics these observed distributions. We also note that the farther that the sender is from the adversary, the more likely that the adversary will observe naturally occurring frame corruption, which should make it more difficult to distinguish naturally occurring corruption from artificial corruption.

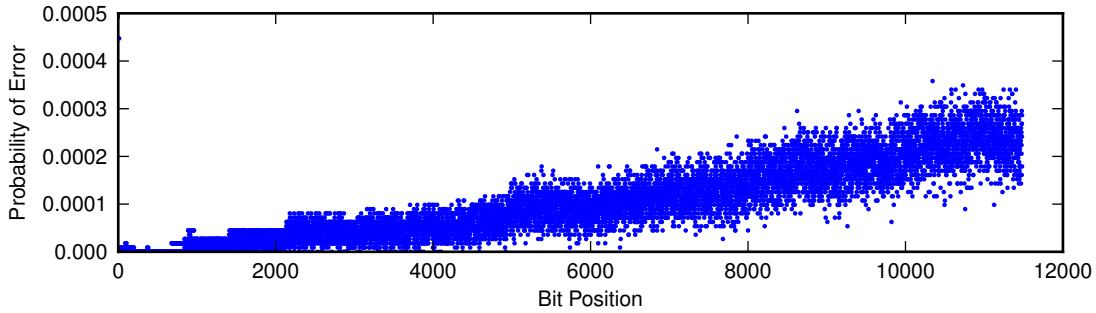
4.5.2 Security Goal

The security of *DenaLi* requires that: (1) sending a hidden message using *DenaLi* creates a perturbation of the wireless channel's packet error rate and bit error distribution that is statistically indistinguishable from if a *DenaLi* message had not been sent (*deniability*); (2) the adversary cannot recover the messages (*confidentiality*). As the confidentiality of *DenaLi* relies on the strength of existing encryption technologies, we focus on defining and evaluating *DenaLi*'s deniability properties.

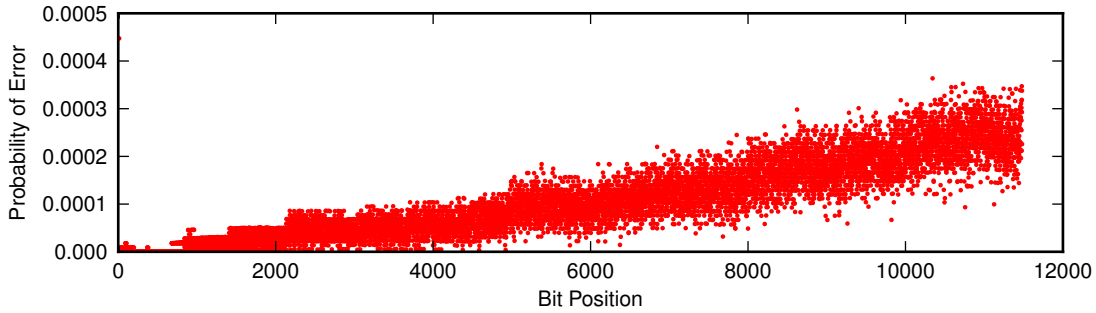
Consider an adversary who observes the properties of the wireless channel from a par-



(a) The bit-error distribution from the perspective of the *DenaLi* sender, given a 23 KB message and a 70-byte TUN MTU.



(b) Natural bit error distribution.



(c) The bit error distribution after the *DenaLi* perturbation from (a) is added.

Figure 4.7: Bit-error distribution in an injected *DenaLi* frame at the sender, and bit error distributions as viewed at a monitor, with and without injected *DenaLi* frames.

ticular location. The adversary can empirically measure both the packet error rate for a sequence of frames, and the bit error distributions within each corrupted frame. Suppose that the adversary has two packet traces P and P', where P is a packet trace without *DenaLi* communication and P' is a trace with *DenaLi* communication. *Deniability* says that the adversary cannot determine which trace has *DenaLi* communication with probability greater than $1/2 + \epsilon$. If the adversary can correctly detect the presence of a covert channel

with probability greater than $1/2 + \epsilon$, then the adversary wins.

Similarly, suppose also that the adversary runs a maximum likelihood detector based on observations of bit error distributions in corrupted frames to detect the presence of a *DenaLi* channel based on deviations in the respective distributions. According to the definition of deniability above, if ϵ is zero, the best threshold that an adversary could design would be unable to distinguish the two distributions of bit error patterns drawn from P and P'. The ϵ parameter measures the extent to which the two distributions do not overlap. We quantify the degree to which the two distributions do not overlap (which corresponds to the probability that the adversary succeeds) using the Pearson correlation coefficient between the two distributions [60]. ϵ is simply half times one minus the correlation coefficient. Formally, we denote the bit error distribution from packet trace P' as $f'(x)$, where x is the bit position in the packet; similarly, the normal bit error distribution from packet trace P is $f(x)$. For each of the distributions that are parameterized by frame error rate and bytes injected per frame, we compare the two distributions as follows:

$$\epsilon = 1/2 - \frac{cov(f(x), f'(x))}{2\sigma_{f(x)}\sigma_{f'(x)}}$$

Note that we can make ϵ arbitrarily small: If *DenaLi* injects no bits from the hidden message, the naturally occurring bit error distribution is unperturbed, and the two distributions are indistinguishable, both by definition and by construction. Such a channel, of course, is useless because its throughput is zero. Increasing the throughput of the hidden channel by injecting additional corrupted frames and introducing bit errors that deviate from the naturally occurring bit errors perturbs the underlying distribution. Thus, there is a trade-off between the degree to which the bit error distribution is perturbed (*i.e.*, the number of bits from the hidden message that we inject into any corrupted frame) and the resulting throughput.

The packet error rate also has a naturally occurring value that varies over time. Suppose that for a given time interval i in packet trace P, the adversary observes a packet error rate

f_i. Then, the adversary can observe a distribution $F = \{f_1, f_2, \ldots, f_n\}$ and a corresponding distribution F' for packet trace P'. We say that the packet error rate induced by running *DenaLi* achieves deniability if the adversary cannot succeed in distinguishing F and F' with a probability greater than $1/2 + \epsilon$. By defining ϵ according to the distance between these two distributions, we can determine the number of corrupted packets that a *DenaLi* sender can inject subject to an upper bound on ϵ. In principle, a *DenaLi* sender can detect the average packet error rate for some time interval and transmit corrupted packets in a way that tracks this packet error rate within some bound of ϵ. For the purposes of our evaluation, we have fixed the packet error rate, but in practice it might vary. Because packet corruption is a local phenomenon that is erratic and unpredictable, fine-grained control over this statistic may not be necessary or useful in practice.

4.5.3 Evaluating Deniability vs. Throughput

In this section, we evaluate the tradeoff between deniability and throughput of the *DenaLi* channel using our prototype implementation. We first describe the experimental setup and then present the results.

Experimental setup

We design an experiment with a sender, a receiver, and a single adversary. Each device is a laptop, where the sender and receiver are configured as described in Section 4.4. The sender generates cover traffic by browsing Gmail over a secure HTTP connection. The adversary is a third laptop with a wireless interface card configured in monitor mode. We locate the adversary in close proximity to the receiver, which, as we described in Section 4.2, is the place where the adversary has the highest probability of detection. We assume that the adversary has only a single monitor. Each node in the setup collects packet traces and records the corresponding packet error rates and bit error distributions, allowing us to see these statistics at the sender, receiver, and the adversary.

Results

We first study the bit-error distributions that result from injecting chunks of hidden messages for a 70-byte MTU for the TUN device. Next, we study how this injected error distribution looks when viewed from the adversary, modeled as a monitor located near the receiver. Finally, to measure how throughput varies with deniability, we explore the relationship between the throughput of the *DenaLi* channel and the Pearson correlation coefficient between the normal bit error distribution and perturbed bit error distribution as seen at the adversary and the resulting throughput of the hidden message corresponding for the corresponding perturbation.

Figure 4.7a shows the bit-error distribution that results from injecting about 23 KB of a hidden message across a sequence of wireless frames, assuming a 70-byte MTU for the TUN device. We choose this size for the TUN MTU because previous studies [50] have shown that about 75% of corrupted packets have bit errors that are less than 400 bits, and a 70-byte MTU and 256-bit HMAC corrupts at most 100 bytes.

Figure 4.7b shows the original bit error distribution for chaff traffic, as viewed from the monitor; Figure 4.7c shows a similar distribution *after DenaLi* has injected a hidden message; as the figures show, the two distributions are essentially indistinguishable. For such a configuration, given "chaff" traffic throughput of about 2 Mbps and a packet error rate of every thousandth packet, *DenaLi* achieves a hidden message rate of about 6 bps. Although the two bit error distributions are not identical, they are reasonably close to one other. The Pearson correlation coefficient between these two distributions was 0.99801, yielding an ϵ value on the order of 10^{-4}. Part of the reason that the two distributions are so close is the relatively low throughput that we have chosen for the *DenaLi* channel. In the rest of this section, we further explore the tradeoff between the level of deniability that the *DenaLi* channel provides and the throughput that it achieves.

Our goal in the first experiment was to demonstrate *DenaLi*'s ability to achieve deniability with respect to bit error distributions. We control the frequency and the extent of

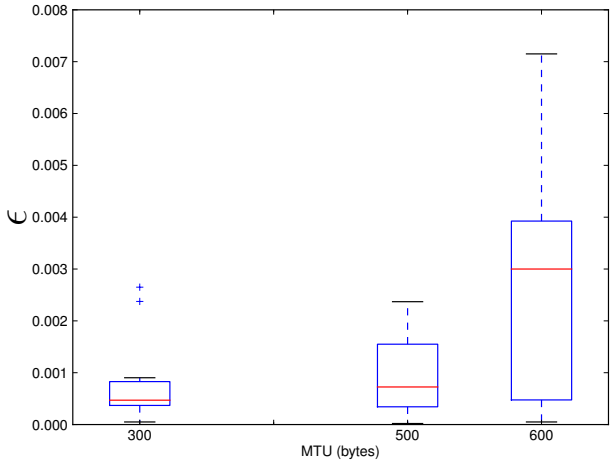


Figure 4.8: ϵ vs. TUN MTU (*i.e.*, injected frame size). We varied MTU sizes to achieve different throughput. Large TUN MTU values result in larger ϵ values and are less deniable.

Table 4.1: Bit error rates, approximate corresponding packet error rates assuming 1500-byte packets, and the resulting *DenaLi* throughput given a 70-byte TUN MTU. We test a range of bit error rates that are observed in practice [61].

BER	**PER**	**Throughput (bps)**
10^{-4}	0.7	427.4
10^{-5}	0.1	103.6
10^{-6}	0.05	42.98

corruption so that the corruption is natural. Because the channel may further corrupt bits in the frame, we are conservative in how we corrupt bits in the frame, which naturally restricts throughput. We now explore how a sender can achieve higher throughput in exchange for less deniability (*i.e.*, a larger ϵ value). We inject one packet for every 10,000 frames of cover traffic This packet injection rate which clearly limits the maximum throughput we can achieve to (at most) 0.0001 of the throughput of the cover traffic, making the two distribution indistinguishable to the adversary. Figure 4.8 shows how ϵ varies as we increase the TUN MTU (i.e., throughput of the *DenaLi* channel). Naturally, ϵ increases with MTU. The throughput of the *DenaLi* channel is also directly proportional to both the throughput at which the chaff traffic is being sent and the packet error rate.

Finally, we study how the throughput of the *DenaLi* channel varies as we vary the packet error rate. To explore a range of packet error rates, we draw from the range of bit error rates reported in the PPR study [61] and convert these observed rates to the corresponding packet error rates in this operating regime. For this experiment, we fix the MTU of the TUN interface to 70 bytes and send SSL chaff traffic by uploading a large file to a Gmail server while varying the packet injection rate (*i.e.*, the rate at which we inject corrupted frames containing hidden messages). Note that fixing the packet error rate and the MTU size is a rough mechanism for controlling ϵ, since the deviation is controlled by the size of the *DenaLi* block size (*i.e.*, the TUN MTU). We then measure the corresponding throughput (which is directly proportional to the throughput of the chaff traffic). Table 4.1 shows how the throughput of the *DenaLi* channel varies with packet error rates for a range of operating regimes. The channel efficiency is similar to previous experiments; as expected, the bitrate of the channel increases as the channel noise increases, as a noisier channel affords more opportunities to inject corrupted frames without deviating from "normal" packet corruption profiles. We caution that although traffic rates appear faster, the increase comes at the cost of deniability, as we showed in Figure 4.8.

4.6 Discussion

Here we discuss open issues, including both weaknesses with the current *DenaLi* design and avenues for future research.

Coping with limited wireless bandwidth Our experiments show that the cover traffic overhead for *DenaLi* is anywhere from about 10:1 (for high ϵ) to 100:1 (for low ϵ), depending on the burst of errors introduced and the frequency of they are injected. In any case, the amount of cover traffic required to achieve deniability is significant, and it may be prohibitive in settings where users bear high data-usage costs or face usage caps. Although the overhead of cover traffic is inherently necessary for systems such as *DenaLi*, it may be

more inconvenient for our use cases, where users may be communicating over *DenaLi* on wireless networks that are not that well-provisioned in the first place (*e.g.*, coffee shops, public squares). We intend to conduct more experiments in these types of settings to better understand the tradeoffs between the overhead that typical users would face and the deniability that they would need to achieve.

Analysis of bitrate adaptation algorithms In this paper, we have ignored the topic of bitrate selection. 802.11 devices have multiple bitrates to chose from, and some senders will decrease their bitrate when they encounter poor frame reception rates, hence corrupt frames may be transmitted at different rates than the eventually successfully received copy. In our prototype, we transmit corrupted frames at 1 Mbps. This is due to the limitation of commodity hardware as the chipset does not allow different transmission data rates. An adversary might profile the bitrates of the corrupted frames to discover anomalous bitrate adaptation patterns. The particular fallback rate(s) are determined by algorithm implemented by the driver at the sender, which might be vendor specific. For *softmac* drivers in the Linux distribution, the rate algorithm is Minstrel, and the fallback rates can be configured in the frame's transmit descriptor. This might be a problem if *DenaLi* is deployed in peer-to-peer wireless network and high operation might bring down the overall throughput of the wireless network. We still think future wireless drivers might support all different rates in debug modes.

Timing attacks The adversary could perform more sophisticated timing attacks to discover a sender who is using *DenaLi*. Under ordinary circumstances, when a sender transmits a corrupted frame, the sender should follow that frame with a retransmission and ultimately receive a corresponding link-layer acknowledgment. Our implementation may not give rise to retransmissions within the appropriate time bounds; in particular, in the worst case, an adversary might see the corrupted frame and the retransmission within a very short time interval (possibly even simultaneously). This limitation results because of

DenaLi's implementation on an off-the-shelf wireless chipset which constrain how we can modify the behavior of the wireless MAC. A software radio platform such as Sora [62] could be used to build a system that ensures that duplicate corrupted frames always precede the corresponding non-corrupted frame and link-layer acknowledgment, but such a prototype would not be as immediately deployable as *DenaLi*.

Transport Users can build two-way communication reliability using TCP or application-layer acknowledgements. Denali provides a decoupled virtual interface which gives *DenaLi* users freedom to choose. The attacker can mount DDoS attack by replaying corrupted packet traces, but we can see that *DenaLi* does not have a high overhead on commodity laptops. Also, the underlying encrypted hidden messages might arrive out of order, requiring a transport-layer protocol like TCP.

Smartphones Our current prototype implementation of *DenaLi* was implemented on Linux laptops, but a likely deployment scenario for *DenaLi* might be on smartphones (*e.g.*, where citizens, operatives, or soldiers in a common area are coordinating and may only have small personal devices). In some of these areas, we might expect 802.11 WiFi deployments, in which case porting *DenaLi* to mobile devices might suffice. In some cases, 802.11 access points may not be deployed, in which case deniable communication might need to depend on some other wireless communication medium (*e.g.*, cellular, bluetooth). Current smartphones equipped with Snapdragon processors have clock cycles up to 1 GHz, which is powerful enough to process *DenaLi* packets.

Multi-hop wireless networks *DenaLi* currently operates only where the sender and recipient are within radio range of one another (*i.e.*, typically on the same wireless LAN). Although we believe that there are significant opportunities for using *DenaLi* in these settings, additional deployment opportunities exist in multi-hop wireless mesh networks, many of which are now explicitly being deployed for the express purpose of Internet freedom [63].

In these settings, *DenaLi* might still be used to deniably pass messages between each pair of participants (*i.e.*, it could form the "link layer" anonymous communication protocol), but applying *DenaLi* to a mesh network setting is less straightforward. First, doing so would involve constructing an overlay network of participants to relay the message, where the relays would be chosen both according to the level of trust for each participant, as well as their (rough) geographic location. Participants may also have to re-inject hidden messages into newly corrupted packets at each hop to avoid intersection attacks; doing so is not straightforward, since the intermediate hops may not possess the key to decode the hidden message.

Mobile Adversaries *DenaLi* can be identified by an adversary which can fingerprint different wireless chipsets with hardware differences and tolerances built in them. This would be an analysis at the physical layer/ RF spectrum. This can happen when the adversary is very close to the transmission. We acknowlege such limitation in the current implementation. This can change if the wifi-chipset architecture does not ask the registers values to be burned at compile-time and only one chipset is used which allows to change the value of register at run-time.

Analysis of FEC Current implementation of *DenaLi* injects random FEC value in the link-layer checksum. It does not have control on the errors it can introduce on the 4 bytes of FEC which can be exploited by the adversary to detect it. This is the limitation of current Atheros chipset which generates the FEC while the frame is in transmission over the air. This is to meet timing constraints of the protocol. We think the future implementations might provide to inject a different value with improvements in the speeds of chip circuits.

4.7 Summary

Citizens of the world have an increasing need to achieve private communications in public spaces. Unfortunately, public meetings are observable, and users who are communicating

with one another may need more covert means of exchanging messages when they are in close proximity. In many cases, users may wish to hide the fact that they are communicating in the first place. We suggest that parties who are near one another should take advantage of packet corruption in wireless networks to provide cover for their communications. To do so, we develop *DenaLi*, a lightweight deniable communications system that allows parties to exchange messages in a local setting, without exposing the fact that they are communicating. We take advantage of the ubiquitous nature of 802.11 "WiFi" networks to construct a covert communications channel, using corrupted packets as the "chaff" to hide communications between parties.

We have designed and implemented *DenaLi* using real end hosts and commodity wireless interface cards, demonstrating that such a system is practical. Our experiments explore the tradeoff between the deniability of the communications (*i.e.*, the extent to which the profile of packet corruption matches "normal" corruption characteristics) and the throughput that the user can achieve when sending hidden messages. Like many anonymous communications systems, *DenaLi* requires significant communications overhead in terms of the cover traffic that users must send to achieve deniability.

CHAPTER 5
POWER-LINE WHISPERER

5.1 Introduction

While encrypted communication becomes increasingly commonplace, end-to-end encryption may not be sufficient to ensure complete privacy for certain types of communication. In contrast to cryptographic systems, which aim to protect the confidentiality or integrity of communications, we focus on a different problem: *concealing the presence of communication*. In communication systems that are designed to achieve confidentiality, third party may nonetheless be able to detect and prove the existence of communication, even if it is not possible to decipher the communications. Communications based on Cryptographic primitives operate on restrictions imposed on the computational power of the adversary while relaxing assumptions on the underlying physical layer. It assumes that physical layer is error-free and the communication trace is reliable. In contrast, *information theoretic security* [64, 65, 66, 67, 68] exploits an advantage that is based on properties of the physical layer, where the adversary has access to the signal from which it cannot extract information about the message. The increasing prevalence of side-channel attacks [69, 70, 71, 72, 73] which render cryptographic schemes vulnerable and meta-data leakages [74, 75] that create the need for new methods that provide stronger guarantees at lower layers of the protocol stack.

Our primary goal is to design a *deniable* communication channel—one that conceals even the *existence* of communication—using the ubiquitous power transmission infrastructure, where the channel noise can be used as a cover for covert communication. In this paper, we design and implement a physical-layer deniable communication system called *PowerLine Whisperer*. It is deniable in the sense that it allows participants to plausibly

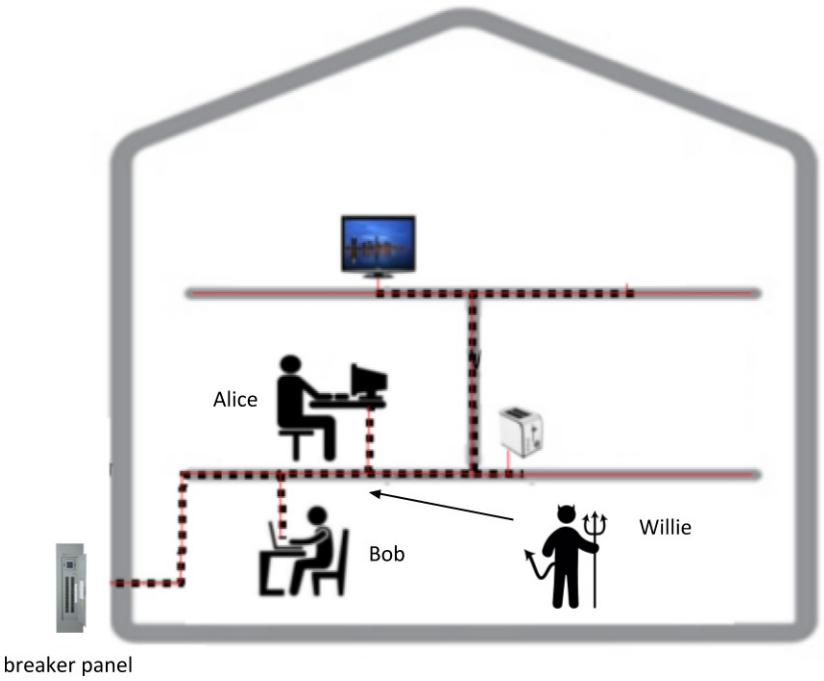


Figure 5.1: Application scenario of a setup where Alice and Bob might use *PowerLine Whisperer* to achieve deniable communication inside a cafe over common powerline circuit. Alice communicates to Bob in-spite of presence of an adversary connected next to him on the same wall power-socket.

deny exchange of information between them making it statistically ambiguous for an adversary to detect the presence of such communication.

Figure 5.1 illustrates a scenario where *PowerLine Whisperer* could be used. It shows a generic meeting place such as a cafe. There are multiple appliances connected to powerline channel, such as coffee machines and toasters, in addition to laptops and other computing devices. Assume a spy in public space, whistleblower in an office setting, or an activist in an authoritarian regime. Alice and Bob would like to communicate deniably; on the other hand, an adversary, Willie, aims to detect the presence of such communication. Powerline networks are isolated at the main breaker panel of the building (usually near the utility meter), and adversary can be anywhere on the powerline. He may tap into a powerline system of a commercial (ex. public spaces like coffee shops), enterprise (ex. business spaces), or residential building at a central location [76], such as the breaker panel or distribution board and deploy sensing technologies to surveil an entire facility. We design *PowerLine*

Whisperer to function in the presence of a strong adversary who is connected to a wall socket next to the parties who wish to communicate.

Because the primary goal of a covert channel is to deny the very presence of a signal on a channel, unless carefully designed often at the expense of complex hardware [32], a guided medium like powerline provides an added degree of covertness that does not require a direct line of sight. The isolation in powerline circuits using circuit breakers acts to our advantage as the eavesdropper requires to be connected to the same building as the suspected parties. Usage of such systems can require massive expenditure from administrative regimes for tapping every wall socket in a building.

Our contributions are as follows. We recognize the need for alternate tools for deniable communication for short range message exchanges. Second, we notice that in urban settings, the ubiquity of powerline network can be a useful communication medium. Third, we notice that the noise on the powerline can be a useful covert cover itself to conceal such communication. Fourth, we define a modulation scheme which achieves deniability on powerline network. Finally, we implement and evaluate a prototype using software radios which can exchange small messages up to $2KB$ in few seconds in a fashion that is indistinguishable from ambient noise on the power line. Although these transmission rates are relatively low, the strong deniability makes this channel appropriate for certain low-volume communication where the communicating parties remain deniable.

5.2 Threat Model

The aim of an adversary is to detect covert communication with high confidence. The threat model borrows the framework from Chapter 3

Basis of Deniable Communication *PowerLine Whisperer* builds on two fundamental observations about communication. Every message at the application layer is encoded into a codeword which provides it reliability on the channel. Error-correcting codes protect the message against channel errors. First observation is that even simple codes (*e.g.*Repetition

codes) use a generator matrix to encode the message, which imposes a structure on the codewords. Secondly, there is always noise on the channel which has a natural variation, which would cause errors on any detector. The transmission of codewords cause disturbance on the channel distorting the distribution of channel. We exploit the above two observations to be deniable on the channel, using the following arguments:

- Key S is used to choose random time slots at which the symbols are transmitted by Alice. This avoids imposing structure on the channel due to time correlations that could be introduced due to the use of different error-correcting codes. Willie does not know the shared key and therefore is oblivious of the times at which information is transmitted.
- The Square Root Law [20] ensures the errors introduced by the natural disturbance in transmission of n bits is $O(\sqrt{n})$ when passing through a noisy channel and the detector will not know whether the $O(\sqrt{n})$ disturbance was caused by transmitter or caused by noise.

$$D_{KL}(Q_0^n||\hat{Q^n}) = \Sigma Q_0^n \log \frac{Q_0^n}{\hat{Q^n}} \tag{5.1}$$

We define the *deniability* of communication as the average probability of errors at the detector as it indicates the uncertainty in the measurement of the adversary, *i.e.*$\gamma = \frac{\alpha+\beta}{2}$. The probability of raising a false alarm is denoted byα, while the probability of misdetection corresponds to β. For a deniable scheme, $\gamma \geq \frac{1}{2} - \frac{\sqrt{D(\hat{Q^n},Q_0^n)}}{2}$. Higher value of γ indicates higher deniability, as it indicates more measurement errors by the adversary. A detector which produces a random outcome, will have a γ close to 0.5.

5.2.1 Detection Strategy

The statistical problem at hand for detection is to decide between two alternative explanation for observations $\lambda(\mathbf{z_w})$, which lies in the domain of hypothesis testing.

$$\lambda(\mathbf{z_w}) = \log \frac{P(\mathbf{z_w}; H_1)}{P(\mathbf{z_w}; H_0)} \tag{5.2}$$

The Neyman-Pearson lemma can be shown to derive the form of optimum test for hypothesis testing [64, Theorem 11.7.1]. This optimum test is of the form of equation 5.2 which is termed as log-likelihood ratio. It represents the probability of observations at Willie under respective hypothesis. The test can be shown to be equivalent to computing the KL Divergence between the true distribution and the observations under the alternate hypothesis as shown by the equation [64, Eq 11.194]. The Lemma seeks a receiver that would maximize the probability of correct detection while keeping the probability of false alarm less than a specified value. It computes the ratio of conditional probability of channel output distributions, given a message was transmitted to the conditional probability of current observation given channel noise, which will take the temporal correlations of the channel output into account.

The transmission vector $\mathbf{x_w} = [x_1, x_2, ...x_i, x_n]$, $x_i \in \{0,1\}$ comprising of information bits travels through the noisy channel. Bit 1 (or bit 0) is represented by a pulse (or absence) of 10 voltage samples $\in \{0,\mathrm{d}\}$. d is a digital value. Transmission of bits affects the channel output $\mathbf{z_w} = [z_1, z_2, ...z_i, z_n]$ of the *matched filter* [77] at Willie, where $z_i \in \{0,\mathrm{d}\}$ before demodulated into a bit stream of $\{0,1\}$. A matched filter matches the received vector to the basis function of the transmitted signal vector. It is a filter with an impulse response $\psi(t) = \phi(T-t), t \in \{0,\mathrm{T}\}$ to a signal $\phi(t)$. We use a matched filter of the transmitted signal

at the receiver because it maximizes the *signal-to-noise ratio* of its output [78, Chapter 4].

$$\lambda(\mathbf{z_w}) = \log \sum_{(x_1,...x_n)} P(z_1,...z_n|x_1,...x_n)P(x_1,...x_n) - \log P(z_1,...z_n|0,...0) \tag{5.3}$$

The first term is the summation over all sets of sequences (with slight abuse of terminology, we can term them as codewords) transmitted for covert message $(x_1,...x_n)$, which contain all possible combinations of O($\sqrt{n}$) information bits out of n. The a-priori probability of each bit transmitted is $O(\frac{1}{\sqrt{n}})$ as only $\sqrt{n}$ of the total n slots will have information bits. To compute the probability $P(x_1,...x_n)$, there are $\binom{n}{\sqrt{n}}$ ways to choose the location of injection of information bits. The total number of terms to be chosen are given by the length of secret key, $\sqrt{n}\log n$ causing the total number of terms to be computed equal to $2^{\sqrt{n}\log n}$. In our experiments, the number of bits on the wire are around $\sqrt{n}$ $(= 1024)$ such that the $\lambda(\mathbf{z_w})$ is computationally prohibitive. As it is computationally hard to compute all posterior probabilities of different sequences transmitted, we compute the amount of disturbance on the channel to detect the presence of covert communication. It is reflected in sufficient test statistic, the variance of the output of the observations of the matched filter. Variance is a sufficient statistic [79] for set of observations sampled from multivariate Gaussian distributions. A statistic $S(z_1,...,z_n)$ is said to be sufficient for parameters θ of the distribution, if the conditional distribution of $z1,...,z_n$ given $S = s$, does not depend on θ for any value of s. The sufficient statistic is important as it summarizes the sequence of observations, without any loss of information.

We use a receiver operating characteristic curve (ROC curve) which is a single graph to capture the performance of detection as one varies the threshold over a range of values of the sufficient test statistic. We plot the probability of detection $(1-\beta)$ versus probability of false alarm (α) to evaluate the deniability of communication. We indicate the optimal point on the curve, which corresponds to the threshold that maximizes the probability of detection by calculating the minimum distance from any point on the ROC curve to the

upper left corner called the *Point of Perfect Classification* as shown in section 5.6.2.

It is important to notice that sometimes noise will cause the decoder to output a digital value 1. The adversary does not have the secret key which is the primary cause of deniability as the adversary cannot be certain that the output is corresponding to noise or an information pulse. We discuss the tradeoff in subsection 5.6.2.

Frequency domain detection One might want to use detectors in frequency domain, in which case it is to be noted that the matched filter projects the received signal into signal space (basis functions) of the transmitted signal. These bases functions are equivalent to Fourier basis in the frequency domain and we are not required to conduct analysis in the dual space as we already use convolution operator in the time domain. Such tests will equivalent to the test conducted in the paper.

5.3 PowerLine Whisperer

In this section, we describe the details of the components and the challenges in realizing the design in practice. We first describe the design parameters followed by description of transmitter and receiver blocks shown in high-level diagram in figure 5.2.

We exercise two design parameters—frequency of transmission, the bandwidth of the channel, and the time over which the message is transmitted—can be modified to balance the tradeoffs between deniability and throughput. We fix the pulse shape used for transmitting the information bit. One can notice a pulse in spectrogram. It is to be noted that claiming the presence of the pulse, however, is not the same as proving the presence of communication. The aim is to use a detection strategy which observes the channel for a period of time and takes a decision as described in subsection 5.2.1.

5.3.1 Transmitter

A message is encoded into a codeword using Reed Solomon code into vector of bits. We pick on of several error-correction schemes [80, 81, 82] have been used on powerline com-

Algorithm 1 Scheme to chose the $O(\sqrt{n})$ slots for transmission of information bits out of total of n slots

1: **procedure** TRANSMISSION SLOT–SELECTION Initialize the random number generator with seed.
2: Let the message size be cM in number of bits. c is a scaling factor; n is the transmission size.
3: **for** $1 \ldots n$ **do**
4: Choose a number from a Bernoulli distribution ($c/\sqrt{n}$) generated by the random number generator
5: Place the message bit in the slot
6: **end for**
7: **end procedure**

Transmit the whole sequence of bits (of size $c\sqrt{n}$) on the channel

munication for the prototype. The choice of the error correcting code does not impact the covertness and is primarily to provide reliability. The encoded message bit sequence is then fed into a Sequence Inflater. The sequence inflater selects the positions generated by Algorithm 1 which is the key S, to place information bits in the long sequence as shown in Figure 5.2), which acts a guiding principle for upper bound on number of transmissions on the channel, reducing the search space in experimental evaluation.

First, we generate random positions for transmission of information bits by using a Bernoulli trials with probability of (c/1024). We use a pseudo random number generator in Linux to produce slots for transmission of message. This forbids us to claim that our system is information-theoretic secure as lack of true randomness might impose a structure on the transmitted sequence (but not as definite as a codeword) which can be exploited by the adversary. The seed used to generate transmission slots can be part of secret agreement between the parties.

The message bits are passed to a pulse interpolator (using root-raised cosine pulse) which generates digital samples to be injected on the channel using Powerline Interface (also called Powerline coupler). The symbols are eventually injected on channel with help of Digital to Analog Convertor. We use non-coherent On-Off Keying (OOK) to transmit messages on the channel which means the pulses do not carry information in the relative

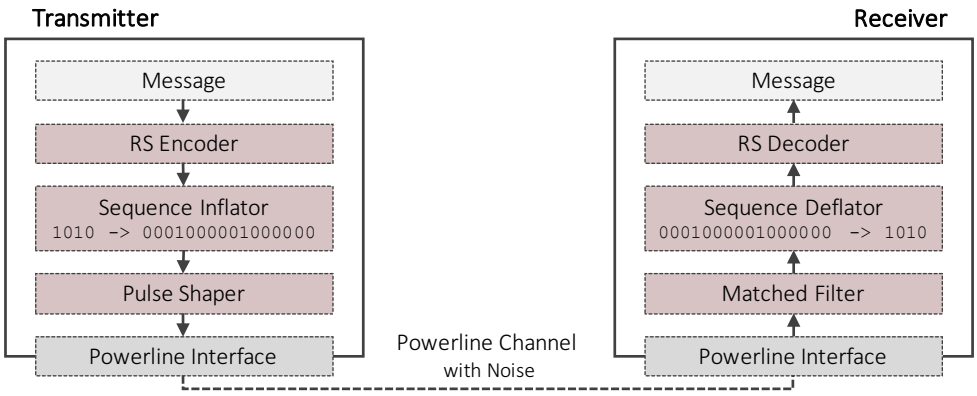


Figure 5.2: High level diagram of Transmitter and Receiver connected to powerline. Some of the building blocks are not shown to preserve clarity of idea and are implementation details.

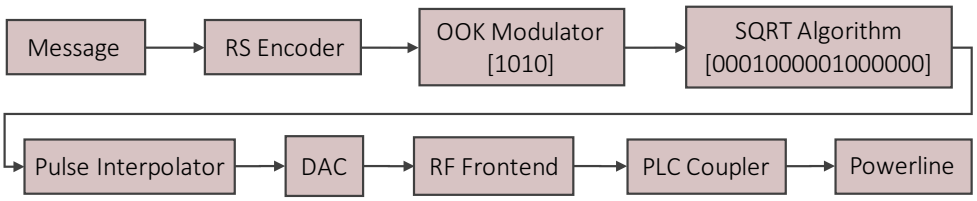


Figure 5.3: High level diagram of blocks in Transmitter chain

phases as they are spread apart in time. This is in contrast to the common modulation schemes (including amplitude modulations) where messages are transmitted in bursts for high throughput and clock recovery.

We transmit a preamble for the receiver to initiate message transfer. The length of the preamble is included in the total transmission length below O($\sqrt{n}$). Transmission of preamble between covert parties can be avoided if they use other out-of-band techniques to derive a common clock between the participants *e.g.* GPS disciplined oscillators.

5.3.2 Receiver

We use a non-coherent receiver that performs envelope detection. It consists of different blocks in the reception chain of a message over the power line. The samples captured from the physical medium using a powerline interface (*i.e.*, a powerline coupler) are passed through the RF front-end to an Analog to Digital Converter. These samples are consumed

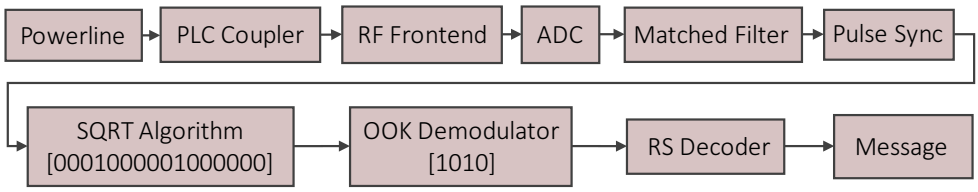


Figure 5.4: High level diagram of blocks in Receiver chain connected preserve.

by a matched filter and uses a *polyphase clock* synchronization algorithm to sample the resulting output. A *threshold* block (both not shown in the diagram for clarity purposes) is calibrated to produce a binary value. The binary sequence which is passed into a sequence deflater that extracts the message bits using the key S. These bits are then passed through an error-correction decoder to obtain a message.

Challenges The messages are converted to a sequence mostly consisting of 0s, allowing the noise between modulated pulses (corresponding to information bit 1) to cause a drift in the receiver clock. This is in contrast to conventional hardware receiver chain, where a scrambler is used to scramble long and randomly varying sequences of bit 0 and bit 1 to enable clock recovery at physical layer, which helps the receiver lock on a frequency. We solve this by increasing the number samples per pulse and increasing the roll-off factor of the pulse.

5.4 Hardware Implementation

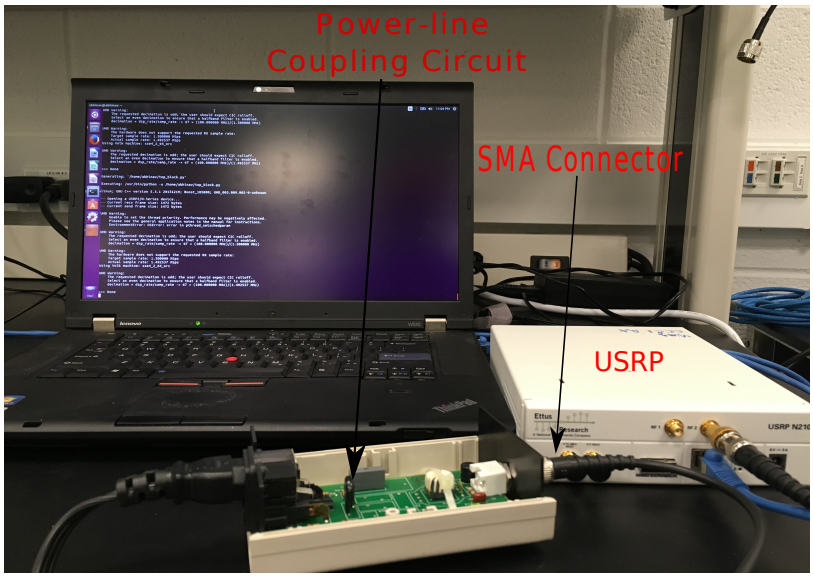
In this section, we detail the hardware used for building the components of that we described in Section 5.3. We have built *PowerLine Whisperer* from simple hardware giving high flexibility. Both of these components can be miniaturized as an oscillator chip and an AC coupler within the power distribution circuit of a laptop or desktop computer for concealment.

Commercial powerline adapters are proprietary design and are not widely available, which makes it difficult to use for covert communication on powerline. These adapters are incapable of covert communication at the physical layer and this work is a leading example

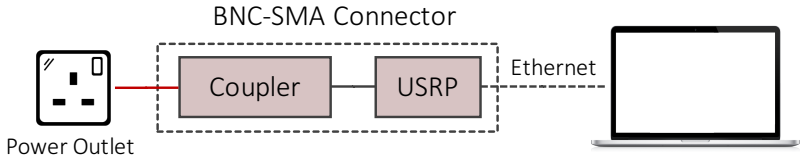
towards benefiting from this untapped potential. Since, there is lack of open-source device-drivers for Ethernet-over-power adapters, we have used Software Defined Radio (SDR), extensively used in the wireless research community for prototyping novel ideas. A Universal Software Radio Peripheral (USRP in Figure 5.5a) acts a generic analog-to-digital and digital to analog (a device with dimensions $22 \times 16 \times 5$ cm) converter which provides reconfiguration flexibility in the receiver and transmitter chain at the cost of its size. GNU Radio [83] is an open-source software development platform for USRP hardware, compatible with different hardware, which can also be used to build the *PowerLine Whisperer*. We also use a powerline coupler, which is a printed circuit board, to transmit and receive electrical signals from the RF front-end of the USRP into the wired powerline on the powerline network.

A stealthy instantiation of *PowerLine Whisperer* could be miniaturized form of the present configuration. (We have built the current prototype proof-of-concept demonstrating the feasibility of the design, rather than as a final version that might be used in deployment.) There are some promising palm-size technologies on the market transceiver like Lime-SDR [84] for frequencies range 100 KHz-3.8GHz and even smaller receivers as RTL-SDR and AirSpy [85, 86] and we think will reduce in form-factor and allow deniable communication on powerlines in near-future.

The setup in Figure 5.5 shows the relative placement of the different blocks used by the two parties and the adversary. The coupler is an analog band pass filter with range of frequencies between 10 kHz to 30 MHz (which are typical frequencies to observe interference caused by electrical devices as mentioned by Gupta *et al.* [76]). It is a 78200-R-PL-20 Line-to-Earth Coupler with line rating of 120 VAC. The coupler is an important component since it is used to couple the signals generated from the USRP daughterboards connected to it over a BNC-SMA cable. We use N210 USRP [87] units for the transmitter, receiver, and passive adversary. The USRPs use LTX/LRX daughterboards to inject and capture the signal. The daughterboards can sample the channel over a frequency range of 0 to 30 MHz,



(a) A power-line coupler PCB (printed circuit board) PCB connected to GNU Radio (ADC/DAC) using SMA cable.



(b) Schematic corresponding to the above image. Describes the relative positions of hardware components.

Figure 5.5: Setup for message injection and collection over the powerline channel.

which encompasses the range of frequency of interest. Sampling is the process of digitizing continuous band pass signals in Digital Signal Processing. The samples captured from powerline refer to the "observations"mentioned in Section 3. We sample the continuous waveforms on the powerline channel to generate digital samples in discrete time domain and then conduct an investigation on the samples to calculate deniability of *PowerLine Whisperer*.

Calibration The USRP daughterboard transmitter front-end has a transmission power of 7dBm. We modify the baseband magnitude to vary the transmitter power. We calibrate the receiver which decides its decoding capability at low signal to noise ratio and apply error correction to retrieve the message using the secret key. The threshold for an information pulse to be decoded as binary value 1 depends on the noise floor. For example, when

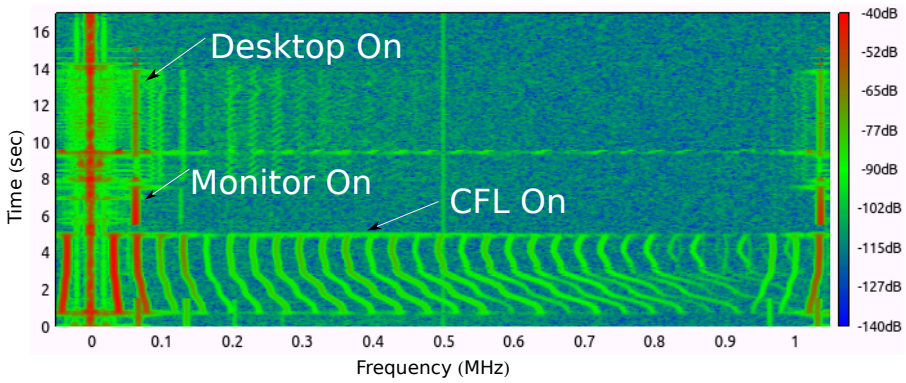


Figure 5.6: Capture of EMI produced by different appliances connected to an isolated transformer for identifying different frequencies of devices for annotation. *PowerLine Whisperer* uses the presence of noise on powerline for deniable communication. The noise present at 500 KHz is an artifact of the daughter board used of signal capture and not due to any device.

the ambient noise floor is higher, the threshold of the receiver matched filter should also be higher to calibrate the device for low false positives. We use the *Threshold* block, to convert the digital output of matched filter into a binary output of bit 0 or 1. *False positives* occur when adversary claims that a bit is transmitted when it is actually not transmitted, instead the noise matches at the instance of the information pulse. It means the adversary has falsely decoded a noise pulse as a binary value 1 at the output of the matched filter. We notice in Figure 5.8a that noise will also produce a significant output at the matched filter. *False negatives* occur when an information bit 0 is transmitted but the adversary's matched filter is not able to detect because it does not have the secret key.

5.5 Primer: Ambient Powerline Noise

In this section we elaborate on our experience with powerline noise. Figure 5.6 shows characteristic noise generated from electrical activities on powerline circuits by the devices such as CFL bulbs, desktop computers, laptops, LCD monitor connected to it, apart from the background noise in an isolated environment a Line Isolation transformer in the lab setting (Tripp Lite 500W isolation transformer). There is presence of various appliances on powerlines such as resistive loads (electric oven, microwave door lights etc) and inductive

loads (mechanically switched dryer, dishwasher etc) which do not produce large electrical noise but they produce thermal noise (johnson noise) and transient noise due to making and breaking of circuits in switches. Loads drawing more than .25 amps produce large transient noise and more prominent continuous electrical noise due to switching mode [88]. We observe the prevalence of noise in the frequency ranges of the powerline (*i.e.* 10 kHz to 30 MHz) and propose that covert communications can be injected in any part of the frequency spectrum in this range, with sufficient noise. Gupta [76] *et al.* built on the assumption that noise on powerline channel can be modeled as additive white Gaussian noise (AWGN) at lower frequencies. Our experience show that the noise is colored and can be a mixture of Gaussians. This means that the power spectral density of powerline channel varies with frequency. Small frequency bands can be found to have almost constant power spectral density. The lower frequencies within the range of few kiloHertz have high noise floor depending on the presence of number of devices in use while the higher frequencies contain the harmonics.

The powerlines can be made from different types of cables and with different periods of installation causing differences. Our experiments have suggested that all devices might not show strong presence of characteristic frequency(switching frequency of the converter) unless they are using SMPS converters including certain light bulbs (excluding CFL). We have found that enterprise buildings have multiple independent powerline circuits, some of which are heavily loaded with devices while others are relatively clean and have a trend with time of day. The parameter space for *PowerLine Whisperer* is vast as it can operate on vast range of frequencies, at different times of day and depending on the density of appliances and users that it is difficult to draw a baseline characterization of the channel in different practical settings. This makes the adversary's job harder to detect covert communication as he does not have a baseline of noise for comparison and aids in providing deniability to the users.

One can observe the modalities in the noise in frequency domain and use the opera-

tional state of various devices using a Gaussian function (mean and variance) to communicate. There is an inherent variability in the noise generated by devices due to the tolerance in components. This nature of noise can also be a cause of deniability. Changing the operational state of the device to modulate bits is interesting but most of the devices are not programmable and would require to be modify their configuration manually by user. We discuss a programmable device (e.g. laptop in section 5.7. Instead of restricting to noise generated by specific device at specific frequency, we allow a generic system to transmit on different frequencies with background noise as message cover.

The Square Root Law holds for Additive White Gaussian noise and the powerline noise might be multivariate Gaussian in amplitude and might not be strictly white. We can decompose the colored spectrum into smaller frequency ranges of constant power spectral density (white) and apply the law.

5.6 Evaluation

In this section we answer the following questions in our empirical evaluation of the *PowerLine Whisperer* prototype:

1. To what extent is communication deniable?

2. What throughput does *PowerLine Whisperer* provide?

We first conduct real world experiments followed by controlled experiment suggesting the limits on throughput of the scheme.

5.6.1 Experiment Setup

We use our hardware setup and pre-select parameters such as frequency and bandwidth of transmission for our evaluations, as we discuss below. We do our experiments at different frequencies of 1.8 MHz, 2.6 MHz, 3.5 MHz in enterprise, commercial and residential environments. We choose different frequencies as they provide high noise cover for message

exchange. The transmitter uses a 200 KHz signal to transmit covert messages while the adversary over-samples the signal by factor of 10.

The operation regime of *PowerLine Whisperer* depends on a number of factors from he nature and activity on the channel governing the noise characteristic to the sensitivity of the adversary. In our experiments, we transmit over a total number of slots of $n = 1024 * 1024$, with time for one slot being the same as time to transmit one bit. For a constant $\sqrt{n}$, we change transmission size linearly from factor (c=) from 3 to 15 for different messages sizes (detailed description in subsection 5.6.3.)

We model our adversary connected to a single vantage point, and claim that presence of adversary at multiple vantage points is not helpful in the joint analysis of the traces as the signal and noise will attenuate with line impedance (which is function of distance) between the point of sensing and the point of transmission. We allow the adversary to plug its measurement device in the wall socket next to the transmitter and operate in the same frequency range electrical voltages using USRP and conducts measurements at the same frequency as the transmitter in our experiments. The adversary is oblivious of the frequency and bandwidth of signal and captures a wider spectrum of 2 MHz. Practically, communicating parties can make the frequency and transmission bandwidth a part of the secret agreement needed to bootstrap the communication, hence the leeway available to hide messages is actually much higher, as is the difficulty for the adversary to detect the communication. If the adversary knows the exact bandwidth of operation, it reduces the throughput of the covert parties although still allow communication to take place. In practical scenario, the colluding parties can always move to less adversarial powerline socket.

Measurements We conduct several trials on the powerline channel with equal number of the experiments conducted in presence of the covert communication as the number of experiments in the absence of any communication aiding to establish the ground truth. The experiments are conducted for the same period of 5.2 seconds which is constant across all experiments. We refer transmission of one bit on the wire as an instance corresponding one

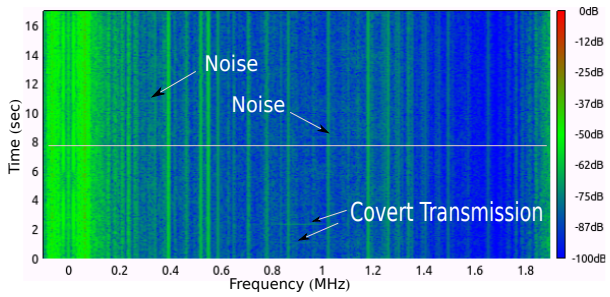


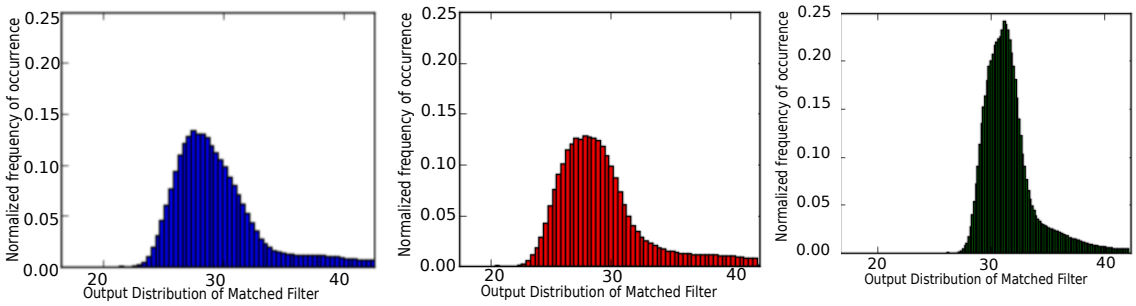
Figure 5.7: Figure shows the 2 MHz band spectrum centered at 900 kHz showing covert transmission in enterprise setting. The top half of the channel shows a time period when there is no message injection. The lower half of the channel shows *PowerLine Whisperer* in operation. Powerline has different frequencies where noise looks similar to the covert message. It looks innocuous to the naked eye. We further investigate using fundamental ground of statistical testing.

channel use out of n.

5.6.2 Deniability of Communication

Figure 5.7 shows a spectrogram of enterprise powerline channel where the top half is the channel without transmission and the lower half shows covert transmissions. It is possible to see a pulse on a spectrogram, which seems innocuous. One can use a different pulse shape similar to noise, which will give different tradeoffs in evaluation. It is important to understand that claiming the presence of a pulse, however, is not the same as proving the presence of communication. The aim is to use a detection strategy which observes the channel for a period of time and takes a decision as described in subsection 5.2.1. The reason for adversary's inability to distinguish between noise and signal pulses on average is because the KL divergence between the two distributions is negligible. KL divergence is the measure of the adversary's ability to distinguish, and we use framework of hypothesis testing to evaluate it.

Figure 5.8 shows transmissions captured at the adversary, identifying the change in the distribution of the output of matched filter in residential setting. The variance in case of noise 5.8a is 76.38, in the presence of covert communication 5.8b is of $2KB$ is 70.06 by the covert transmitter. The variance changes to 30.15 when there is a transmission of



(a) Channel profile in absence of covert message injection (channel noise).

(b) Channel profile after injection of covert message, showing non-visible perturbation.

(c) Channel profile after injection of continuous pulse train(shifted mean and variance).

Figure 5.8: The three sub-plots show normalized histograms of output of sufficient statistic at 3.5 MHz transmission frequency and 2 MHz bandwidth in residential setting. The figure demonstrate the channel output in extreme conditions and the fact that the covert communication will resembles noise profile.

constant train pulses by the covert transmitter in a pathological case of high throughput overt communication 5.8c. The fundamental idea for covert transmission is to be close to the original distribution of noise by sparse transmissions over a period of time.

Figure 5.9 shows why it is challenging for an adversary to detect communication with *PowerLine Whisperer*. We consider a set of 200 experimental trials measured at the adversary, half of which contained covert communication. In this case, suppose the adversary uses a threshold on the sufficient statistic, that is, the variance of the distribution output matched filter to determine the presence of covert communication.

The blue curve (with star points) in Figure 5.9 shows the empirical probability that a trial with covert communication present is detected by the adversary, as the threshold on the variance of the sufficient statistic is varied. Notice that by choosing a low value of the threshold (left half of the graph), it is possible to detect almost all covert transmissions. However, the adversary would also detect several trials which did *not* have covert communication. We show this in the red curve (with dot points), which captures the empirical probability that a trial actually contains a covert communication given that it is detected by the adversary.

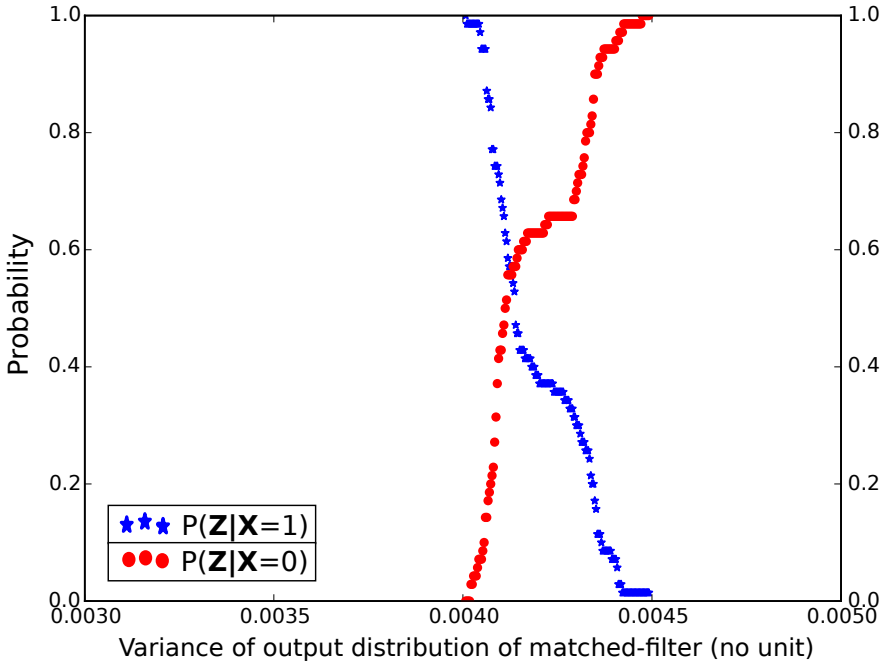


Figure 5.9: The tradeoff at the adversary in qualifying a detection as covert communication or noise with variation in the value of threshold at the detector. The threshold is the variance of the output of the matched filter. Figure shows how the variation in the Left y-axis shows True Positive $(1-\beta)$ and right y-axis shows False Positives (α).

The situation on the right half of this graph is the opposite. At the cross-over between the two curves, the adversary would be unable to distinguish a detection of covert communication from an actual covert communication with more than a (roughly) 50% chance. This is an inherent tradeoff present in any detector. The best operating point corresponds to a point where the probability of detection is the greatest, given the lowest probability of error, which is evaluated using ROC curve.

The ROC curve in Figure 5.10 shows the variation in deniability on powerline in an enterprise setting. Every point on the curve describes the false positive rate and true positive rate for a threshold value at the adversary. We show 95% confidence intervals in the rate of detection at a steady change of threshold values. The tight error bounds show that measurements are meaningful. We observed tight confidence intervals for false positive rates, but we chose not to plot them to provide visual clarity to users. γ_w denotes the deniability index for communication corresponding to an optimum point of operation for the adversary.

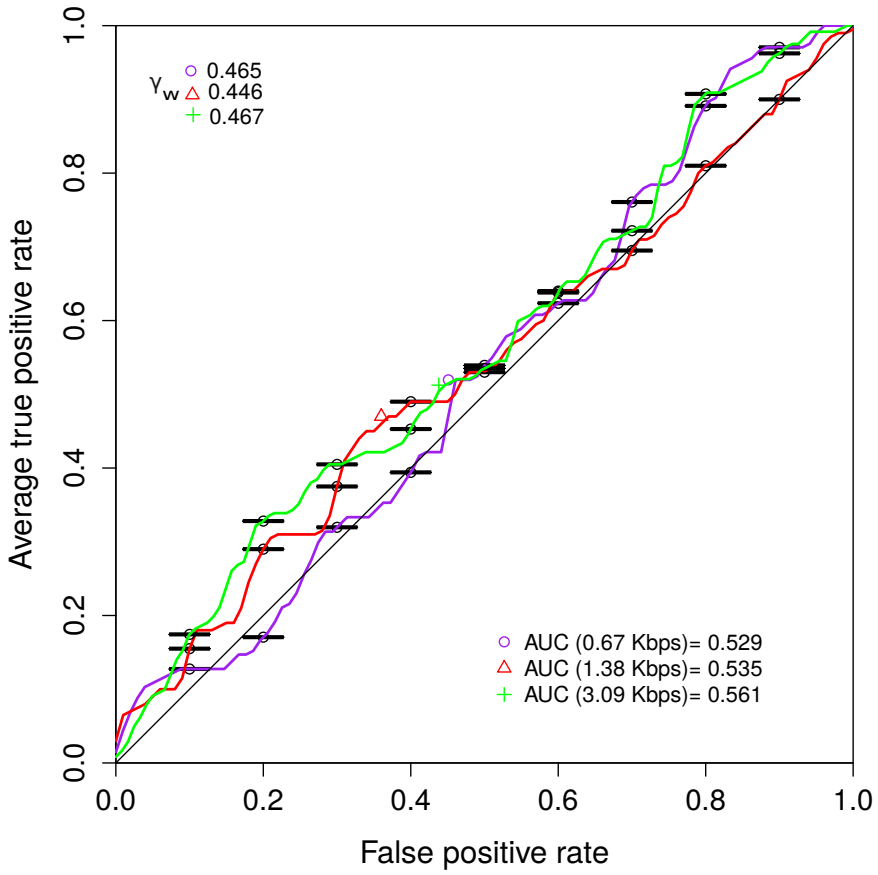


Figure 5.10: The plot shows ROC curve at the adversary with change in message size in an enterprise setting. The sender operates at 1.8 MHz.

They are at a minimum distance from the point of perfect classification, corresponding to a threshold that gives the maximum probability of detection or minimizes the probability of error of the detector, corresponding to Neyman-Pearson Lemma discussed in section 5.2 (having the least error detection probability) for statistical hypothesis testing under the two distributions.

This is the worst possible threshold for point of view of the colluding parties for each case of message throughput and shows one can be deniable during message exchange. We get a deniability of greater than 0.4 most experiments. This number itself encodes the information about how good or bad is the detection with respect to noise (as we conduct an equal number of trials for observing each hypothesis.) It suggests one can make error in

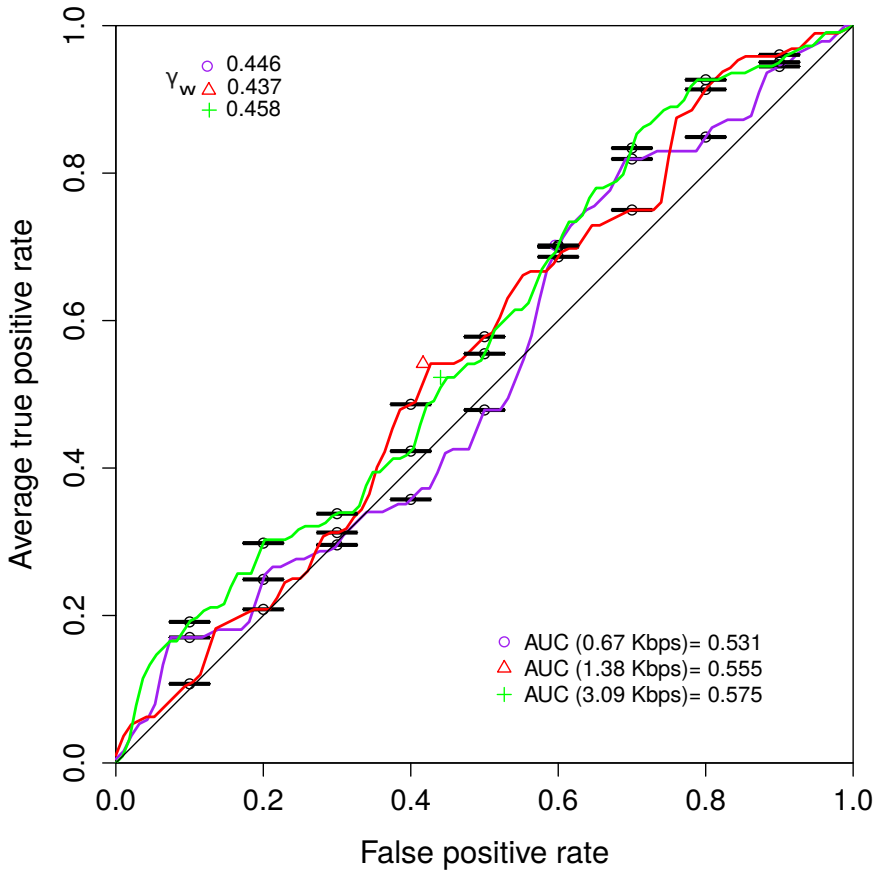


Figure 5.11: The plot shows ROC curve at the adversary with change in message size in a residential setting. The sender operates at 3.5 MHz

observation 4 out of 10 times to detect the covert communication (close to half the time.)

The ROC curve in Figure 5.11 shows the variation in deniability in a residential setting. The actual receiver of the message can always decode the message in these experiments. The value of area under the curve being close to 0.5 suggests the ROC curve of the detector is close to the line of no discrimination ($1 - \beta = \alpha$). It shows the low accuracy of Log-likelihood ratio test (near-optimal test) which detects as many true positives ($1 - \beta$) as the number of false positives α.

The ROC curve in Figure 5.12 shows the performance of the detector in a commercial environment of a student cafe. The deniability offered in cafe assumes to be seemingly less than other environments. The is due to the absence of enough noise cover as there were

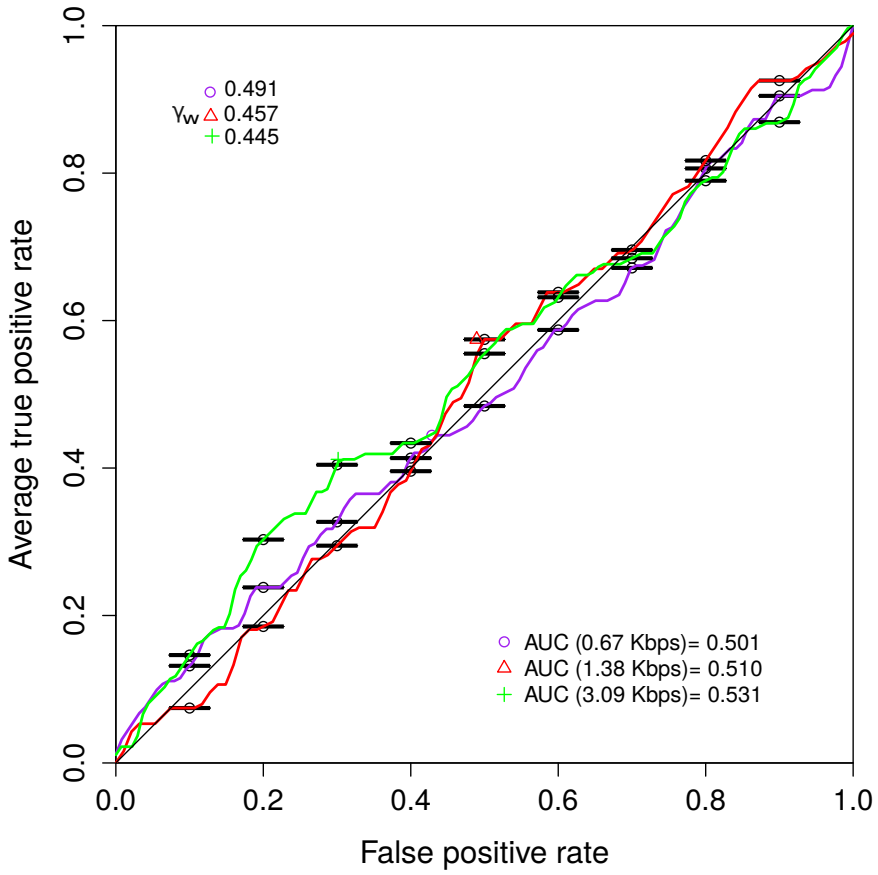


Figure 5.12: The plot shows ROC curve at the adversary with change in message size in a commercial setting. The sender operates at 2.6 MHz.

hardly any students around when the experiments were conducted in the late hours of the day. We later show trends in how the noise power changes during the evening hours in the enterprise setting.

5.6.3 Throughput

The physical layer on the power line does not have a link layer protocol in our setting, allowing transmission of bits allowing us to transmit at the rate we inject data on the channel, without conventional link-layer back-offs. We defined throughput as the rate we inject the message into the channel. Short messaging might range from hundreds of bytes (eg. "tweets") to larger message sizes. We use message sizes of 432, 896, 2008 bytes to be

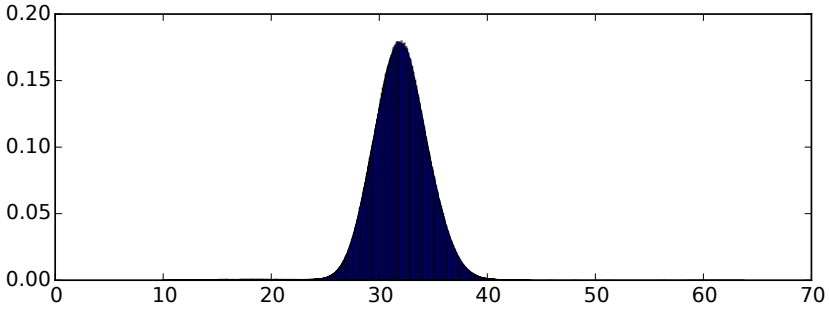
Table 5.1: We summarize the results of experiments showing the variation of Area under Curve with bit-rate achieved in different environmental conditions.

Bitrate (Kbps)	**3.08**	**1.37**	**0.67**
Commercial (2.6 MHz)	0.501	0.510	0.531
Residential (3.5 MHz)	0.531	0.555	0.575
Enterprise (1.8 MHz)	0.529	0.535	0.561

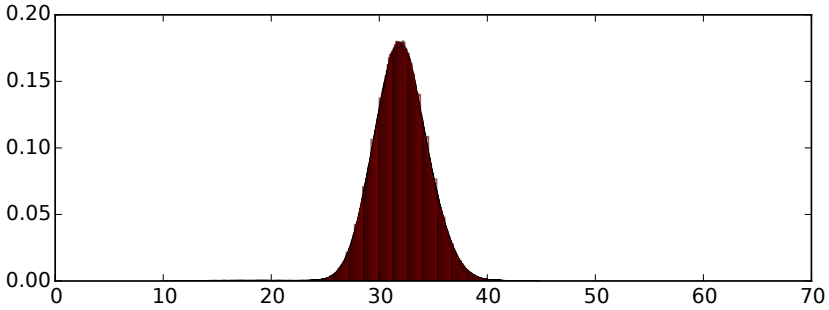
transmitted over the channel over a constant period of 5.2 seconds. We ran experiments for a single message over several runs but one can transmit messages over sustained period of time.

Table 5.1 shows the variation of the Area Under Curve (AUC) with the bit-rate achieved in commercial, residential, and enterprise settings. The table shows that increasing the bit-rate reduces the deniability of *PowerLine Whisperer*. Figure 5.13 shows the similarity in the channel distribution of 150 experiment runs each during the presence of only noise and covert message transfer. The slight difference in the shape can be equally attributed to channel noise or to message transfer, giving the inherent deniability to leverage noise at the physical layer to user's advantage.

In our measurements in different environments, we found 22 devices in enterprise, 8 devices in home and 3 in cafe. The devices can be at different distances from wall-socket and might not contribute to noise as much as the devices plugged near. Since the operating conditions in different environments vary due to the number of devices and the characteristic of noise generated by each of them, nature of the background noise due to type and lifetime of powerline cables, it is difficult to compare the detection accuracy of the detector with number of devices. We conduct controlled experiments to observe the trend in Figure 5.14 in the accuracy of detector by injecting 2 KB message on a separate transformer with increasing the number of devices from 1 to 4. We find that the accuracy of the detector decreases with the number of devices as expected. Since it is not feasible to model signature of every device in different environments and account for variability of each of them while they perform different tasks (section on SMPS noise) and cancel the noise generated,



(a) Histogram of 150 experiment trials of powerline channel with channel noise in an enterprise setup.



(b) Histogram of 150 experimental observations of powerline channel with covert message injection in an enterprise setup.

Figure 5.13: The histograms of the measurements in the presence and absence of measurement show non-significant difference providing deniability to the users.

we conduct 200 control experiments to test the scheme on an isolated powerline using an isolation transformer at 1 MHz. We achieve a throughput of 146 bps for deniability index of 0.43. We think *PowerLine Whisperer* can provide a meaningful communication throughput in real settings with increase in noise.

We understand that the more knowledge an adversary has about the underlying distribution, the more difficult it is to produce a higher throughput deniable communication channel. We think *PowerLine Whisperer* can provide a meaningful communication throughput in real settings as the chances for unpredictability increases.

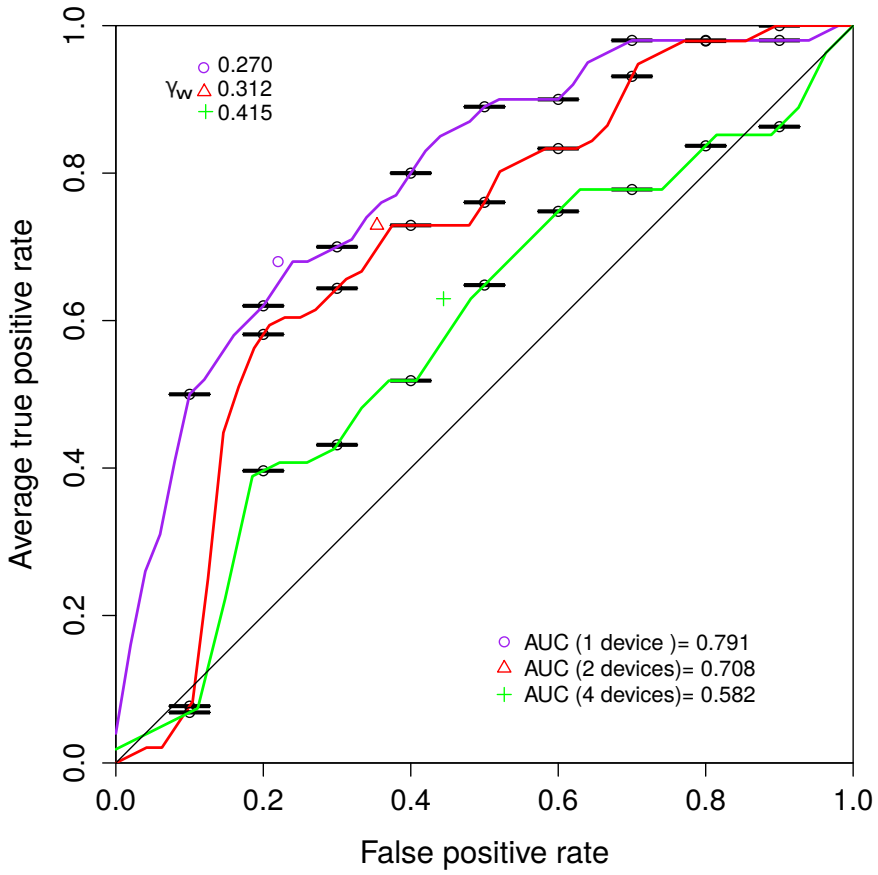
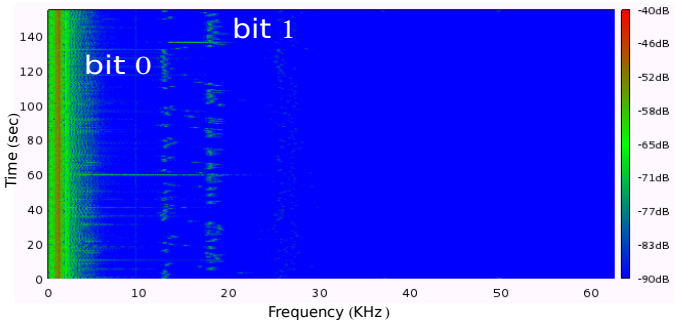


Figure 5.14: The plot shows ROC curve at the decrease in the accuracy of the adversary when the number of devices are increased for constant message size.

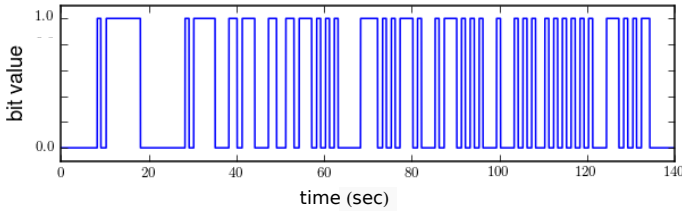
5.7 Discussion and Future Work

Pulse shape We can design different pulse shape and corresponding receivers, for the purpose of different degree of covertness. We can interpolate noise pulse shapes of different appliances and design a pulse shape using empirical noise on a specific channel in *PowerLine Whisperer*. This might be appealing to the human eye and one might claim it is a better covert communication scheme, but an optimal detector (or near-optimal detector used) will analyze the system in a manner exactly as described in the paper.

SMPS Noise We can build a covert channel using SMPS noise produced by the powered devices like laptop adapter and modulate them programmatically. We have conducted ex-



(a) Spectrogram (4096 pt) for a signal generated by SMPS convertor inside a laptop adapter sampled at 125 Ksamples/sec.



(b) Received bits messages using FSK demodulation

Figure 5.15: Deniable Communication using SMPS noise

periments with different models of laptops and other devices such as TP-Link router and found this phenomenon is present in almost all computing devices. Figure 5.15a shows message transmission due to variation computation load on a laptop and Figure 5.15b shows the demodulated using Frequency Shift Keying (FSK). This scheme will also require software-defined radio to capture signals from powerline with a receiver chain using a matched filter (with pulse of shape of average noise bursts from the adapter) generated by the laptop adapter. Such a scheme is interesting but restricted to frequencies around the switching frequency of the SMPS converter of a device, restricting a degree of freedom in *PowerLine Whisperer*. Deniability in such cases will require one to model the normal workload of such devices. Frequency modulation of such scheme is slow due to the analog circuitry (capacitance and inductance) and produces low throughput for an adversary model described in the paper.

Connection to Networking Stack Our current scheme demonstrates a working prototype on a single transmission link and as follow up, we are interested how to extend the scheme

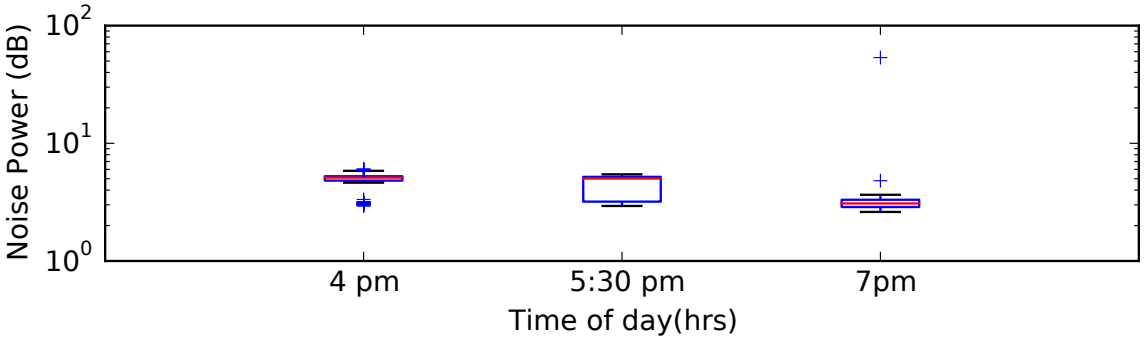


Figure 5.16: Change in noise power(variance) with time of day. Each box-plot is for observations every 20 seconds over the previous 1.5 hours. Possibly indicating people leaving for home in the evening.

over multiple grids and cities. We are also interested in how we can interplay our scheme with services on the Internet and the infrastructure required to build it. It also requires multi-user MAC protocol and interfacing it with a transport protocol such as UDP for large message transfers.

Applications in Air Gap Systems Assume an enterprise organization's Ethernet network equipped with different levels of access rights; different users in the same building might have machines that are connected to different switch ports and accordingly assigned to different VLANs, as a means of isolating their communication from one another. The powerline can be used to covertly bridge these "gaps". Since the users are deniable, they cannot be strictly held responsible for using the channel. It enables communication across multiple rooms within a building that are not otherwise connected by Ethernet (*i.e.*, an "air gap").

Time of day effects There is natural variability present in power-line networks, which might vary due to several reasons including human activity, weather or power faults in the power-line. Such effects, although variable can have a periodicity at different scales spanning a few days to the time of the day as captured in Figure 5.16. This can give an added advantage to the users if they have the knowledge of frequency ranges which are noisier and more suitable for deniable communication due to activity on power-line.

Connecting Power-line Whisperer to Tor

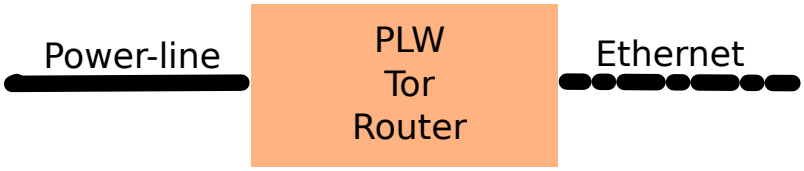


Figure 5.17: Connecting *PowerLine Whisperer* to Tor Network

It might be interesting to connect *PowerLine Whisperer* to Tor where one can add hardware box (as shown in Figure 5.17) which converts messages on Power-line messages to Ethernet which can then be relayed on the Internet. This can be used for deniable or anonymous message boards although it might be tough to maintain TCP connection at low connection speed. These portals might be public where people can go and plug in their laptop to send messages. This solution is better against an Internet-Wide Adversary than a local adversary for two reasons. First, the Internet-Wide global adversary does not have any idea about local power-line networks. Second, boxes (power-line to Ethernet converter) can be as small as Raspberry Pi and be readily deployed like Onion routers.

Jamming the channel *PowerLine Whisperer* cannot communicate if an adversary can jam the entire channel, although it seems impractical. The users may choose to operate in a different frequency band as powerline offers a wide range of frequencies (order of GHz). A practical adversary might want to inject pulses in between the transmission sequence. Such an adversary can best be evaded by using better error correcting codes.

Random Number Generator We have thought about and understand that the random number generator provided by Linux kernel does not have all properties to generate the random numbers to claim the scheme to be Information theoretic secure but it does provide something practical to be used. The effect of not using a perfect random number generator is that it will introduce a structure on the transmitted sequence of bits which will not appear completely random to the adversary.

Extensions to Multi-user settings It would be possible to extend such message exchange

in practical settings, but a congestion of multiple users at the same frequency and time might increase the overall energy on the channel which would raise suspicion in a restrictive setting but might be possible in practical settings.

5.8 Summary

We built *PowerLine Whisperer*, first of its kind physical-layer deniable channel on power-line against a strong adversarial model having rigorous theoretical grounding. It is flexible in operation and simple in design, providing selective use of the channel, opening exciting directions to use power-line channel for covert communication. It uses ubiquitous power-line to cleverly hide messages in abundant noise.

CHAPTER 6

CONCLUSION

In conclusion, I will reiterate the point that there is no single solution that can provide deniability in every setting – wide area or near field. In near-fields, one would like to compare the performance of *DenaLi* and *PowerLine Whisperer*.

As one would have noticed, these two systems operate at two different layers of the networking stack. Although the cause of corruption of bits is noise, the noise profile would be different in each case. One can also notice that the Wifi and Power-line have different channel models. The modulation scheme (Phase-Shift Keying) in *DenaLi* was used for spectral efficiency than for covertness, while the modulation scheme (On-Off Keying) in *PowerLine Whisperer* is suited for the purpose of providing covertness.

To compare the bitrates, we should be able to compare the noise in common-metric. In Wifi, quantification of noise in link layer is represented by errors in different positions of bits we compare the noise using the distribution over bit error rate. In Power-line, we compare the distribution of sufficient statistic over the output of matched-filter. In the former case, it is computed over distribution after hard-decoding of bits while in the latter case, the distribution is over computer over voltages before hard decoding of bits. The definition of deniability can be derived from the equation $\alpha + \beta = 1 - V(\hat{Q}^n, Q_0^n)$, where Q_0^n, $\hat{Q}^n$ are the distribution of channel in the absence and presence of covert communication and α, β are the probabilities of type I, II errors. In *DenaLi*, we use the right-hand side to derive the value of deniability by comparing the two distribution using correlation coefficient to test the linearity in the distribution. In *PowerLine Whisperer*, we recognize the limitation of estimating the error distribution under different hypothesis and instead calculate the probabilities of error in detection. The detectors used are similar in nature, but they are different in operation. They are similar as they are computing the error (doubt of adversary) but it

is different as they compute different sides of the equation. The placement of adversary and the capabilities of the adversary are not the same. In *DenaLi*, the user can improve the deniable communication throughput by increasing the cover traffic, while *PowerLine Whisperer* is dependent on underlying channel noise distribution which is not in control of the user to be perturbed to his advantage. A 6 bps rate for 2 Mbps cover traffic against an adversary operating at link layer is different from 146 bps against an adversary operating at physical layer as the threat models are different. A higher value of throughput might not be a direct conclusion about which system is better since the way the base distribution of noise appearing at the two layers is different (and difficult to be put in a common metric). There is a difference in the computation of deniability metric in the two cases, which suggests that the number might not be an absolute value that can be compared across two systems.

6.1 Summary of contributions

This thesis has demonstrated that one can do deniable communication. In doing so, we made the following contributions:

1. *Two systems for conducting and measuring deniability of communication. DenaLi* and *PowerLine Whisperer* are two systems which are useful in doing deniable communication using two different mediums - Wireless and Power-line.

2. *Technique for deniable message transfer.* We explored how private communication in the future might be protected by *DenaLi*. *DenaLi*'s security properties will be invaluable against future attackers, which could be used by people in pairs in close proximity. To explore beyond the limitations imposed by a wireless ASIC, we used Software-defined radios to conduct finer measurements using *PowerLine Whisperer*. In doing so we explored a much deeper body of the covert communication channel using information theoretic modeling which provides fundamental limitations on such systems.

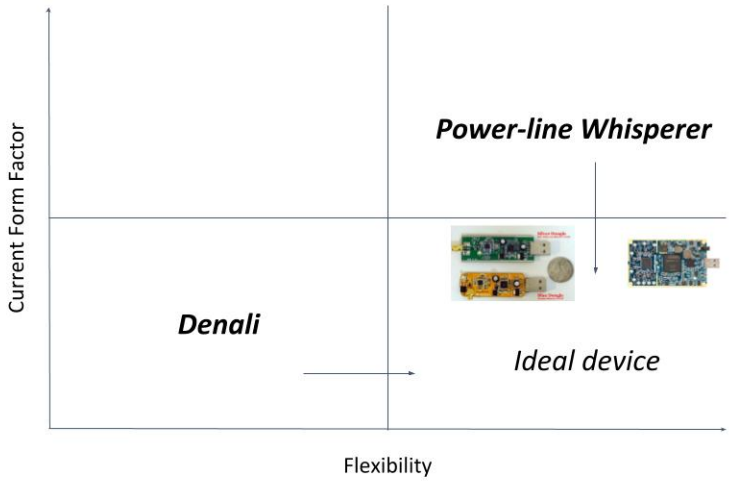


Figure 6.1: Future directions for near-field tools for Deniable communication

6.2 Future work

We conclude by summarizing promising research to be done in building deniable communication systems. I have attempted to build point-to-point deniable communication systems during my research. Much of the interest has been in finding out if such systems can become reality in the first place.

As wireless communication is improving with the advent of new protocols (802.11 n, ac), there are differences how the channel models are used by the new schemes. There are more antennas used for spatial diversity and the mechanism for transmissions whereby more than one packet might be transmitted in the air simultaneously are challenging to use the 802.11 a,b,g specifications. The newer protocols change how the packets are queued in the buffers in the device drivers, which in turn can cause a vulnerability in implementations of the idea of using corrupted frames in wireless networks. Although the user can always choose to not use the latest protocols for communication.

Some of the vendors have open-sourced their drivers with contributions from Qualcomm Atheros and Broadcom, others have not while they continue to deal differently with their wireless transmission (Broadcom or Intel). Discussing powerful ideas of deniability

out for discussion and explaining the purpose of such schemes might motivate the companies to be open to provide certain functionalities in their commercial chipsets, which will immensely increase the adoption of such mechanisms in the future.

Reducing the form factor of a system like *PowerLine Whisperer*, might be the biggest challenge. While there is an obvious adoption of wireless technology for communication, it is not the case with powerlines. We haven't been able to find an open-sourced Ethernet-over-power card but using such schemes over commercial products can be advantageous for deniable communication in the future. There are software-defined radios coming up on the market with reduced form-factor [84] and it will be interesting to use *PowerLine Whisperer* on it. As technology evolves, we can move towards the lower right quadrant of Figure 6.1 where we can find a reasonable tradeoff at an operational point between the form-factor and the flexibility allowing deniability on the channel as well as in physical appearance to users.

6.3 Applying techniques to other problems

Power-line networks might have an untapped potential in context of securing devices in home and away. Internet of Things has proliferated the commercial market in recent years with exciting use of technology augmenting human experience. Rapid deployment of IoT devices has not left enough time for developing security solutions leaving millions of devices as ticking time-bombs on the Internet being exploited by hackers in recent past. Power-line is an essential backbone that can be used to maneuver around this potential threat by leveraging information leakage from these computing devices. We further the pursuit of information leakage from power-lines which uses the noise generated by SMPS transformers to detect anomalous activities by a computing device.

The Internet of things eco-system is vulnerable to security threats from hackers around the globe. There are many IoT devices online and there are no tools to check the health of these devices. We attempt to un-obstrusively detect security breaches in IoT devices be-

fore and report it to users with the devices installed using a mobile application. Figure 5.6 shows electromagnetic interference generated by devices on power-line channel. These emanations are primarily caused due to SMPS (Switched Mode Power Supply) [89]. Current solutions where one device(sensor) monitoring consumed power by the device can be very limiting, we try to look for centralized approach where one could use the interference produced on power-line to reveal anomalous behavior on a computing device. Nature of such activities can vary depending on the kind of devices and can be evaluated using different micro-benchmarks. First, Radiated mode where AC power chord is an efficient antenna with length quarter of wavelength for the RFI frequencies present in digital equipment and switching power supplies. Second is the conducted mode. This is further characterized into two modes - Common mode (asymmetrical) where RFI is present on both the line and neutral current paths with reference to the ground or earth path. Differential mode (symmetrical) RFI is present as a voltage between the line and neutral leads.

Computing devices run at the rate of few gigahertz clock frequency and instructions last over few nanoseconds. The accurate regeneration of the instructions requires high sampling rate > 50 Gsamples/sec are very expensive (order of few hundred thousand dollars). Modeling general purpose computer state space is infeasible and hence we have adopted a black-box approach, which enables us to focus only on signals that are consistent and causal with different activities. At best one can get coarse representation of instructions executed on devices using inexpensive equipment. We find side-channel analysis of power-line characteristics of devices an interesting direction to investigate the EMI signatures of software operating on device. In this work tries, we perform coarse grained analysis of the activities of IoT devices. Such an analysis can be done non-intrusively by connecting the system into existing power-line channel. Our approach is non-invasive as it does not require one to open the device chassis. We thus think, it provides a new attack vector to the current approaches which can be complimentary to network traffic analysis.

6.3.1 Experimental Observations

We conducted experiments on the following set of devices listed in a table 6.1. We installed a Line Isolation transformer in the lab setting (Tripp Lite 500W isolation transformer), which isolates the system from most of EMI present on the power-line, so that we can clearly observe the EMI generated by our micro benchmarks. All our results are based on this independent installation setup. The devices run the scheduled OS processes as on usual system boot. Figure 4.1 describes the physical setup of connecting a computing device next to the data collection setup to extract electrical voltage samples from power-line using GnuRadio [83] which uses a 14 bit ADC.

TP-Link Router	OpenWrt
Raspberry Pi 3	Debian
Amcrest View Camera	Proprietary
Nest Camera	Propreitary
Amazon Echo	Proprietary

Table 6.1: Devices used and the operating system running on them

We found adapters of some devices like TP-Link router, Raspberry Pi and amcrest view camera emit EMI on the channel reflecting the state of device. The following spectrograms visually represent the different EMI generated by Raspberry Pi.

Due to lack of original malware samples running on devices, we ran a client *cpuminer* to mine crypto-currency Litecoin on Raspberry Pi on an isolated power-line. This is to emulate the workload of an IoT botnet called *Linux.Darlloz*. Figure 6.2 shows the EMI generated by the process while mining crypto-currency.

Similarly, we configure a client binary of Mirai botnet [90] on Raspberry Pi and compare the EMI signature of a deniale-of-service attack with normal network activity of software updates on the device and we can observe remarkable difference in the EMI generated.

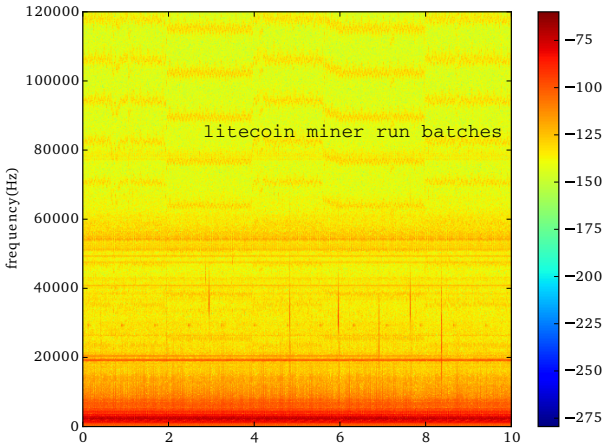


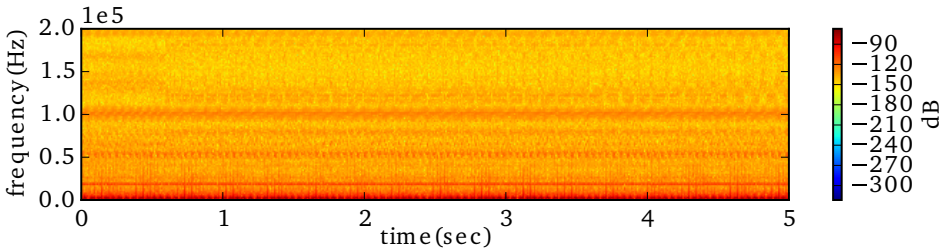
Figure 6.2: Spectrogram of a crypto-currency (litecoin) mining on Raspberry Pi using mining client *cpuminer*

Unfortunately, we were not able to visually observe remarkable differences in the EMI generated by the device correlated with device activity in Amazon Echo and Google Nest Camera. This might be due to the shielding of the EMI generated by the shielding filters on the adapter.

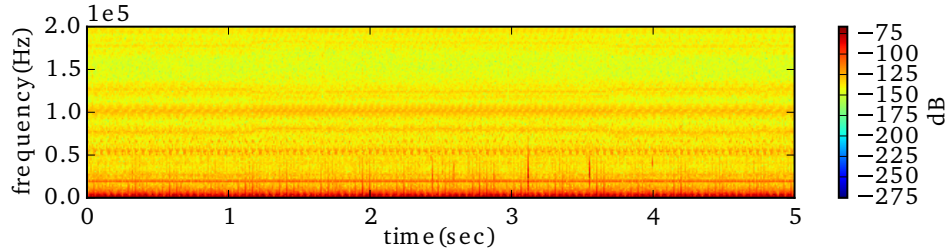
Due to lack of time, we were unable to use beamforming to separate the noise generated from specific device from the noise generated by other devices and I think this is a very interesting area for future research. One can envision building a small prototype which can be plugged into home powerline network to detect infected devices. There might be high computation cost for processing the data generated from sampling from powerline but one can use GPUs as they have become main compute engines for todays high performance computing [91, 92, 93, 94, 95, 96].

6.3.2 Limitation

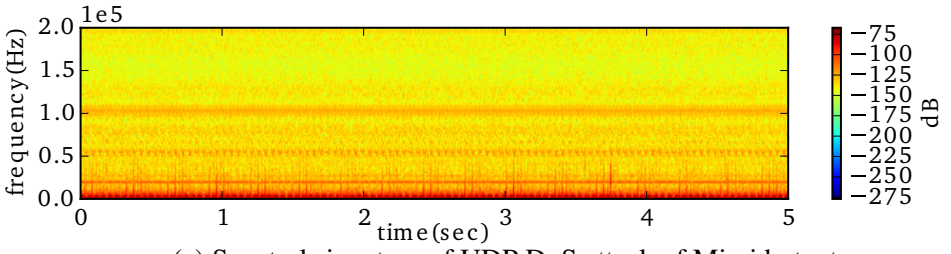
There are certain limitations of the approach because of which we are unable to observe the state of devices such as Amazon Echo or Google Nest because the adapters might be shielding the EMI generated by the transformer in the device adapters. This is expected behavior of the device which follows regulations on the amount of EMI that is generated



(a) Spectral signature of package update (apt-get install texlive-base)



(b) Spectral signature of operating system update (apt-get update)



(c) Spectral signature of UDP DoS attack of Mirai botnet

Figure 6.3: Spectral signatures of part of traces of different activities on Raspberry Pi

by it. Due to the usage of a proprietary hardware which allows us to sample from the channel, we are unable to verify the frequency response of the analog filter. It is unclear that how much improvement in the hardware coupler can provide an insight into what can be inferred from the channel EMI.

6.3.3 Take Away

We see the signatures of different activities on power-line in controlled setting and there is a causation between the activity on the device and the EMI generated on the power-line. Experiments conducted on live power-line on university infrastructure has shown immense interference from the devices already present on the channel which overpowers the noise generated by Raspberry Pi and router adapters. This leaves a lot room for further investi-

gation on how to extract signal from the channel noise, given someone better hardware is used.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *USENIX Security Symposium*, San Diego, CA, 2004, pp. 303–320.

[2] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal, "Raptor: routing attacks on privacy in tor," in *24th USENIX Security Symposium (USENIX Security 15)*, Washington, D.C.: USENIX Association, Aug. 2015, pp. 271–286, ISBN: 978-1-931971-232.

[3] P. Winter, T. Pulls, and J. Fuss, "ScrambleSuit: a polymorphic network protocol to circumvent censorship," in *Workshop on Privacy in the Electronic Society*, ACM, 2013.

[4] H. Mohajeri Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, "Skypemorph: protocol obfuscation for tor bridges," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, (Raleigh, North Carolina, USA), ser. CCS '12, New York, NY, USA: ACM, 2012, pp. 97–108, ISBN: 978-1-4503-1651-4.

[5] F. Harris, *Let's Assume the System Is Synchronized*, R. Prasad, S. Dixit, R. van Nee, and T. Ojanpera, Eds. Dordrecht: Springer Netherlands, 2011, pp. 311–325, ISBN: 978-94-007-0107-6.

[6] G. Danezis and C. Diaz, "A survey of anonymous communication channels," Tech. Rep., 2008.

[7] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, (San Diego, CA), ser. SSYM'04, Berkeley, CA, USA: USENIX Association, 2004, pp. 21–21.

[8] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman, "Telex: Anticensorship in the Network Infrastructure," in *USENIX Security Symposium*, San Francisco, CA, Aug. 2011.

[9] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing," *Communications of the ACM*, vol. 42, no. 2, pp. 39–41, Feb. 1999.

[10] torproject, *Obfusproxy*, Jun. 2016.

[11] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh, "Stegotorus: a camouflage proxy for the tor anonymity system," in *Pro-*

ceedings of the 2012 ACM conference on Computer and communications security, ACM, 2012, pp. 109–120.

[12] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, "Vuvuzela: scalable private messaging resistant to traffic analysis," in *Proceedings of the 25th Symposium on Operating Systems Principles*, (Monterey, California), ser. SOSP '15, New York, NY, USA: ACM, 2015, pp. 137–152, ISBN: 978-1-4503-3834-9.

[13] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. R. Karger, "Infranet: circumventing web censorship and surveillance.," in *USENIX Security Symposium*, San Francisco, CA, Aug. 2002, pp. 247–262.

[14] S. Burnett, N. Feamster, and S. Vempala, "Chipping Away at Censorship Firewalls with User-Generated Content," in *USENIX Security Symposium*, Washington, DC, Aug. 2010, pp. 463–468.

[15] A. Houmansadr, G. T. Nguyen, M. Caesar, and N. Borisov, "Cirripede: Circumvention Infrastructure Using Router Redirection with Plausible Deniability," in *ACM Conference on Computer and Communications Security (CCS)*, Chicago, IL, Nov. 2011, pp. 187–200.

[16] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer, "Decoy Routing: Toward Unblockable Internet Communication," in *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, Apr. 2011.

[17] M. Rogers and E. Saitta, "Secure Communication over Diverse Transports," in *ACM Workshop on Privacy in the Electronic Society (WPES)*, (Raleigh, NC), New York, NY, USA, Oct. 2012, pp. 75–80, ISBN: 978-1-4503-1663-7.

[18] B. A. Bash, D. Goeckel, D. Towsley, and S. Guha, "Hiding information in noise: fundamental limits of covert wireless communication," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 26–31, 2015.

[19] B. A. Bash, D. Goeckel, S. Guha, and D. Towsley, "Hiding information in noise: fundamental limits of covert wireless communication," *CoRR*, vol. abs/1506.00066, 2015.

[20] D. T. Boulat A. Bash Dennis Goeckel, "Square root law for communication with low probability of detection on AWGN channels," *CoRR*, vol. abs/1202.6423, 2012.

[21] B. A. Bash, A. H. Gheorghe, M. Patel, J. L. Habif, D. Goeckel, D. Towsley, and S. Guha, "Quantum-secure covert communication on bosonic channels," *Nature communications*, vol. 6, 2015.

[22] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE Journal on selected areas in communications*, vol. 16, no. 4, pp. 474–481, 1998.

[23] J. Fridrich, *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.

[24] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*, 2007.

[25] Y. Wang and P. Moulin, "Perfectly secure steganography: capacity, error exponents, and code constructions," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2706–2722, 2008.

[26] S. Kullback, *Information Theory and Statistics*. New York, NY: Wiley, 1959.

[27] R. L. Pickholtz, D. L. Schilling, Laurence, B. Milstein, and S. Member, "Theory of spread spectrum communicationsa tutorial," *IEEE Transactions on Communications*, vol. 30, pp. 855–884, 1982.

[28] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread spectrum communications handbook*. McGraw-Hill New York, 1994, vol. 2.

[29] H. Liu and G. Xu, *Methods for channel estimation and signal detection of cdma signals*, US Patent 5,905,721, 1999.

[30] G. Burel, C. Bouder, and O. Berder, "Detection of direct sequence spread spectrum transmissions without prior knowledge," in *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, IEEE, vol. 1, 2001, pp. 236–239.

[31] X. Li and H. H. Fan, "Direct blind multiuser detection for cdma in multipath without channel estimation," *IEEE Transactions on signal processing*, vol. 49, no. 1, pp. 63–73, 2001.

[32] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "Secret agent radio: covert communication through dirty constellations," in *International Workshop on Information Hiding*, Springer, 2012, pp. 160–175.

[33] M. G. Kuhn, "Electromagnetic eavesdropping risks of flat-panel displays," in *International Workshop on Privacy Enhancing Technologies*, Springer, 2004, pp. 88–107.

[34] W. van Eck, "Electromagnetic radiation from video display units: an eavesdropping risk?" *Comput. Secur.*, vol. 4, no. 4, pp. 269–286, Dec. 1985.

[35] M. Guri, O. Hasson, G. Kedma, and Y. Elovici, "Visisploit: an optical covert-channel to leak data through an air-gap," *arXiv preprint arXiv:1607.03946*, 2016.

[36] zdnet, *S.c.s.o.a.i.w. malware*, `http://www.zdnet.com/article/amazon-surveillance-cameras-infected-with-malware/`, Accessed: 2017-4-31.

[37] J. Loughry and D. A. Umphress, "Information leakage from optical emanations," *ACM Trans. Inf. Syst. Secur.*, vol. 5, no. 3, pp. 262–289, Aug. 2002.

[38] M. Guri, M. Monitz, and Y. Elovici, "Usbee: air-gap covert-channel via electromagnetic emission from USB," *CoRR*, vol. abs/1608.08397, 2016.

[39] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici, "Bitwhisper: covert signaling channel between air-gapped computers using thermal manipulations," in *2015 IEEE 28th Computer Security Foundations Symposium*, Jul. 2015, pp. 276–289.

[40] M. Guri, Y. A. Solewicz, A. Daidakulov, and Y. Elovici, "Diskfiltration: data exfiltration from speakerless air-gapped computers via covert hard drive noise," *CoRR*, vol. abs/1608.03431, 2016.

[41] M. Guri, G. Kedma, A. Kachlon, and Y. Elovici, "Airhopper: bridging the air-gap between isolated networks and mobile phones using radio frequencies," in *Malicious and Unwanted Software: The Americas (MALWARE), 2014 9th International Conference on*, IEEE, 2014, pp. 58–67.

[42] A. Najafizadeh, R. Liscano, M. V. Martin, P. Mason, and M. Salmanian, "Challenges in the Implementation and Simulation for Wireless Side-Channel based on Intentionally Corrupted FCS," *Procedia Computer Science*, vol. 5, pp. 165–172, 2011.

[43] K. Szczypiorski, "HICCUPS: Hidden Communication System for Corrupted Networks," in *International Multi-Conference on Advanced Computer Systems*, Oct. 2003, pp. 31–40.

[44] T. E. Calhoun, X. Cao, Y. Li, and R. Beyah, "An 802.11 MAC Layer Covert Channel," *Wireless Communications and Mobile Computing*, vol. 12, no. 5, pp. 393–405, Apr. 2012.

[45] S. Chen, M. Setta, X. Chen, and C. Parini, "Ultra wideband powerline communication (plc) above 30 mhz.," *IET Communications*, vol. 3, no. 10, pp. 1587–1596, 2009.

[46] A. Pinomaa, H. Baumgartner, J. Ahola, and A. Kosonen, "Utilization of software-defined radio in power line communication between motor and frequency converter,"

in *Power Line Communications and Its Applications (ISPLC), 2010 IEEE International Symposium on*, Mar. 2010, pp. 172–177.

[47] *Qualcomm powerline product*, `https://www.qualcomm.com/products/powerline`, Accessed: 2017-12-5.

[48] E. Lehmann and J. Romano, *Testing Statistical Hypotheses*. Springer, 2005.

[49] R. L. Rivest *et al.*, "Chaffing and Winnowing: Confidentiality Without Encryption," *CryptoBytes (RSA Laboratories)*, vol. 4, no. 1, pp. 12–17, 1998.

[50] B. Han, A. Schulman, F. Gringoli, N. Spring, B. Bhattacharjee, L. Nava, L. Ji, S. Lee, and R. R. Miller, "Maranello: Practical Partial Packet Recovery for 802.11," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, Apr. 2010, pp. 205–218.

[51] B. Han, L. Ji, S. Lee, B. Bhattacharjee, and R. R. Miller, "All Bits are Not Equal–A Study of IEEE 802.11 Communication Bit Errors," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1602–1610.

[52] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *ACM Wireless Security Workshop (WiSE)*, Los Angeles, CA, Sep. 2006.

[53] D. Turner, S. Savage, and A. C. Snoeren, "On the Empirical Performance of Self-calibrating WiFi Location Systems," in *Proceedings of IEEE Conference on Local Computer Networks (LCN)*, Bonn, Germany, Oct. 2011.

[54] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis," in *ACM SIGCOMM*, vol. 36, Pisa, Italy, Aug. 2006, pp. 39–50.

[55] S. Mathur, W. Trappe, N. B. Mandayam, C. Ye, and A. Reznik, "Radio-Telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel," in *ACM MOBICOM*, San Francisco, CA, Sep. 2008, pp. 128–139.

[56] A. K. Miu, H. Balakrishnan, and C. E. Koksal, "Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks," in *ACM MOBICOM*, Cologne, Germany, Sep. 2005.

[57] *Denali codebase*, `https://github.com/denalidenali/denali`.

[58] *Linux Wireless Drivers for Atheros*, `http://wireless.kernel.org/en/users/Drivers/ath9k`.

[59] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "Softmac-flexible wireless research platform," in *ACM SIGCOMM Workshop on Hot Topics in Networking (HotNets-IV)*, College Park, MD, Nov. 2005.

[60] S. Rolewicz, *Functional Analysis and Control Theory: Linear Systems*, ser. East European Series. Springer, 1987, vol. 29, ISBN: 9789027721860.

[61] K. Jamieson and H. Balakrishnan, "PPR: Partial Packet Recovery for Wireless Networks," in *ACM SIGCOMM*, Kyoto, Japan, Aug. 2007.

[62] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker, "Sora: High Performance Software Radio Using General Purpose Multi-core Processors," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, Apr. 2009, pp. 75–90.

[63] *Commotion Wireless*, `http://commotionwireless.net/`.

[64] T. Cover and J. Thomas, *Elements of Information Theory*, ser. A Wiley-Interscience publication. Wiley, 2006, ISBN: 9780471748816.

[65] I CSISZAR, "Broadcast channels with confidential messages," *Proc. IEEE. Trans. Inf. Theory., 1978*, vol. 24, no. 3, pp. 339–348, 1978.

[66] A. D. Wyner, "The wire-tap channel," *Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[67] S. Katzenbeisser and F. A. Petitcolas, Eds., *Information Hiding Techniques for Steganography and Digital Watermarking*, 1st. Norwood, MA, USA: Artech House, Inc., 2000, ISBN: 1580530354.

[68] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *Image Processing, IEEE Transactions on*, vol. 6, no. 12, pp. 1673–1687, 1997.

[69] A. Bauer, liane Jaulmes, V. Lomn, E. Prouff, and T. Roche, "Side-channel attack against rsa key generation algorithms," in *CHES*, Springer, 2014, pp. 223–241.

[70] Y. Zhou and D. Feng, "Side-channel attacks: ten years after its publication and the impacts on cryptographic module security testing.,"

[71] E. Peeters, "Side-channel cryptanalysis: a brief survey," in *Advanced DPA Theory and Practice*, Springer, 2013, pp. 11–19.

[72] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conference*, Springer, 1996, pp. 104–113.

[73] E. Hess, N. Janssen, B. Meyer, and T. Schütze, "Information leakage attacks against smart card implementations of cryptographic algorithms and countermeasures–a survey," Citeseer.

[74] *Prism program*, https://en.wikipedia.org/wiki/PRISM_(surveillance_program), Accessed: 2017-4-21.

[75] *U.s., british intelligence mining data from nine u.s. internet companies in broad secret program*, http://www.huffingtonpost.com/2013/06/06/intelligence-mining_n_3399025.html, Accessed: 2017-4-21.

[76] S. Gupta, M. S. Reynolds, and S. N. Patel, "Electrisense: single-point sensing using emi for electrical event detection and classification in the home," in *In Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, 2010, pp. 139–148.

[77] G. Turin, "An introduction to matched filters," June, 1960, pp. 311–329.

[78] J. Wozencraft and I. Jacobs, *Principles of Communication Engineering*. Waveland Press, Incorporated, 1990, ISBN: 9780881335545.

[79] R. A. Johnson and D. W. Wichern, Eds., *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988, ISBN: 0-130-41146-9.

[80] T. C. Chuah, "On reed–solomon coding for data communications over power-line channels," *IEEE Transactions on Power Delivery*, vol. 24, no. 2, pp. 614–620, 2009.

[81] R. Hormis, I. Berenguer, and X. Wang, "Baseband transmission on power line channels with ldpc coset codes," in *2006 IEEE International Conference on Communications*, IEEE, vol. 1, 2006, pp. 379–384.

[82] N. Andreadou and F.-N. Pavlidou, "Performance of array codes on power line communications channel," in *Power Line Communications and Its Applications, 2008. ISPLC 2008. IEEE International Symposium on*, IEEE, 2008, pp. 129–134.

[83] E. R. Gnuradio, *Gnuradio*, Jun. 2016.

[84] Micro, *Lime sdr*, https://wiki.myriadrf.org/LimeSDR, Accessed: 2017-4-21.

[85] A. Spy, *Air spy*, http://airspy.com/airspy-r2/, Accessed: 2017-4-21.

[86] RTL-SDR, *Rtl-sdr*, 2016.

[87] Gnuradio, *Gnuradio n210 data-sheet*, 2015.

[88] S. Patel, T. Robertson, J. Kientz, M. Reynolds, and G. Abowd, "At the flick of a switch: detecting and classifying unique electrical events on the residential power line (nominated for the best paper award)," *UbiComp 2007: Ubiquitous Computing*, pp. 271–288, 2007.

[89] L. Rozenblat, *Smps tutorial*, 2017.

[90] T. Crunch, *Mirai botnet*, 2017.

[91] D. Sengupta *et al.*, "Graphreduce: large-scale graph analytics on accelerator-based hpc systems," in *IEEE IPDPSW*, 2015.

[92] D. Sengupta *et al.*, "Scheduling multi-tenant cloud workloads on accelerator-based systems," in *Proc. of*, (New Orleans, Louisana), ser. SC '14, NJ, USA: IEEE Press, 2014, ISBN: 978-1-4799-5500-8.

[93] D. Sengupta, R. Belapure, and K. Schwan, "Multi-tenancy on gpgpu-based servers," in *Proc. of*, (New York, New York, USA), ser. VTDC '13, NY, USA: ACM, 2013, ISBN: 978-1-4503-1985-0.

[94] D. Sengupta, S. L. Song, *et al.*, "Graphreduce: processing large-scale graphs on accelerator-based systems," in *Proceedings of*, (Austin, Texas), ser. SC '15, NY, USA: ACM, 2015, ISBN: 978-1-4503-3723-6.

[95] D. Sengupta *et al.*, "A framework for emulating non-volatile memory systemswith different performance characteristics," in *Proceedings of*, (Austin, Texas, USA), ser. ICPE, NY, USA: ACM, 2015, ISBN: 978-1-4503-3248-4.

[96] ——, "Graphin: an online high performance incremental graph processing framework," in *Euro-Par 2016: Parallel Processing: 22nd International Conference on Parallel and Distributed Computing, Grenoble, France, August 24-26, 2 016, Proceedings*, P.-F. Dutot and D. Trystram, Eds. Cham: Springer International Publishing, 2016, ISBN: 978-3-319-43659-3.